Tight Steady-State Availability Bounds using the Failure Distance Concept

Juan A. Carrasco
Departament d'Enginyeria Electrònica
Universitat Politècnica de Catalunya
Diagonal 647, plta. 9
08028 Barcelona, Spain
juan.a.carrasco@upc.edu

Except for formatting details and the correction of some errata, this version matches exactly the version published with the same title and authors in *Performance Evaluation*, vol. 34, no. 1, 1998, pp. 27–64

Abstract

Continuous-time Markov chains are commonly used for dependability modeling of repairable fault-tolerant computer systems. Realistic models of non-trivial fault-tolerant systems often have very large state spaces. An attractive approach for dealing with the largeness problem is the use of pruning methods with error bounds. Several such methods for computing steady-state availability bounds have been proposed recently. This paper presents a new method which exploits the failure distance concept to bound more efficiently the behavior in the non-generated state space. It is proved that the bounding method gives tighter bounds than previous methods. Numerical analysis shows that the new bounds can be significantly tighter.

Keywords: Repairable fault-tolerant computer systems; Steady-state availability; Continuous-time Markov chains; State space reduction; Bounds

1 Introduction

Modeling plays an important role in the design, analysis and management of fault-tolerant computer systems. These systems are characterized by their stochastic behavior and, accordingly, probabilistic measures are used for their quantitative assessment. Many systems are seen by their users as simply providing service or not. For these systems, dependability measures such as the steady-state availability and reliability are appropriate. The steady-state availability is a useful measure for repairable systems when the long-term behavior is of interest. In some cases, this measure can be computed using combinatorial techniques [1] or closed-product solution queueing networks [14]. However, in general, the dependencies introduced by lack of coverage, failure propagation, operational configurations and maintenance are such that general-purpose, state-level model solution techniques are required. Continuous-time Markov chains (CTMCs) are often used to analyze the steady-state availability of systems with these dependencies and a number of tools for the specification and solution of CTMCs have been developed [2, 6, 9, 13, 15, 22, 25, 27].

The use of CTMC models is hampered by the generally exponential growth of the number of states with the structural complexity of the system. The "largeness" problem has been attacked from three directions[1]: (a) hierarchical model solution [25, 29], (b) state lumping techniques [8, 13, 18], and (c) bounding methods. Hierarchical model solution is possible when the components exhibit independent behavior or have restricted dependencies. State lumping requires the presence of symmetries in the modeled system. Bounding methods compute bounds using detailed knowledge of the model in a subset of generated states G and bounding somehow the behavior in the non-generated state space U. The first such method was developed by Muntz et al. [23] using results from [11, 12]. In the method described in [23] the subset G includes all states with up to K failed components and the behavior in U is bounded by a submodel in which each state represents the subset U_k of U including the states with a given number of failed components k. The bounding submodel includes "forward" transitions upper bounding failure rates and "backward" transitions lower bounding repair rates. A model has to be solved for each "return" state (state of G with exactly K failed components). In order to reduce the computational cost, a state cloning technique is proposed in [23] which modifies the return subset so that it includes the states with $F < K$ failed components. The state cloning technique introduces some looseness in the bounds. Lui and Muntz [20] proposed a refinement of the method for the particular case $F = 0$ by including a reuse technique which, at the price of an additional looseness in the bounds, avoids a complete reapplication of the method each time K is incremented in the search for the desired accuracy. The additional looseness was reduced in another paper from the same authors [21]. Souza e Silva and Ochoa [26] developed state space exploration techniques in which G is generated incrementally following heuristics which try to obtain the tightest possible bounds for a given number of generated states. More recently [5] two algorithms have been proposed which obtain the bounds of the method described in [23] without state cloning, solving $|W| + 2$ and 4 linear systems, where W is the subset of integers k such that G has transitions to U_k.

[1] Simulation with accelerating techniques such as importance sampling [4, 16] is another approach. However, such methods only give a statistical assessment of the accuracy of the solution.

This paper describes a new method to compute bounds for steady-state availability. The method exploits the failure distance concept to bound more efficiently the behavior in U and, thus, obtain tighter bounds. The rest of the paper is organized as follows. Section 2 establishes basic results which: (1) are needed to justify the method, and (2) provide a basis for comparison with the method described in [23]. The bounding method and more specific theoretical developments associated with it are given in Section 3. Section 4 includes implementation details and a summary algorithmic description of the bounding method. In Section 5 we show that the new method achieves bounds which are guaranteed not to be looser than the bounds given by the method described in [23] and compare the computational costs of the new method with the second algorithm given in [5]. Section 6 illustrates the performance of the method using large examples. Section 7 concludes the paper.

2 Preliminaries

We consider models of repairable fault-tolerant systems made up of components which fail and are repaired. Components are grouped in classes, the components of the same class being indistinguishable. Thus, collections of components will be bags. The fault-tolerant system is assumed operational/down as determined by a coherent structure function [1] made up of atoms $c[n]$, which evaluate to true when n instances of component class c are unfailed, connected by logical AND, OR operators. The structure function evaluates to true when the system is operational. In first instance, we will assume all minimal cuts of the structure function, i.e. all minimal bags of components whose failure implies the failure of the system, to be known. We will also show that the bounding method can be easily adapted to cover the case in which only the minimal cuts with up to a given cardinality M are known. A fault-tree whose top event indicates the failure of the system can be easily constructed from the coherent structure function by exchanging the operators AND, OR and substituting the atoms $c[\mathit{inst}(c) - n]$ for $c[n]$, where $\mathit{inst}(c)$ denotes the number of instances of component class c and the new atoms $c[n']$ evaluate to true when at least n' instances of component class c are failed. Efficient existing procedures to compute minimal cuts [19, 24] are easily generalized to deal with bags of events (component failures). We have developed an efficient algorithm [7] for that task. The state of the system is modified by failure and repair events which occur at constant rates, which may depend on the state of the system. Repair events are assumed to involve only one component. Failure events may involve an arbitrary number of components. We will call *failure bag* any bag of components which may fail simultaneously (in a single transition). E denotes the set of failure bags of the model, *FC* the set of different cardinalities of the failure bags of the model, and E_i, $i \in \mathit{FC}$ the set of failure bags with cardinality i. We will assume known: (1) all failure bags of the model, (2) for each failure bag e, an upper bound $\lambda_{\mathrm{ub}}(e)$ for the rate with which the components of e fail simultaneously, and (3) lower bounds $g(k) > 0$, $k > 0$ for the repair rate from any state with k failed components. It will also be assumed that efficient procedures exist for: (1) determining the bag of failed components $F(x)$ in a given state x and (2) determining the failure bag associated with a failure transition. Finally, we assume that there is a single state o with no failed components.

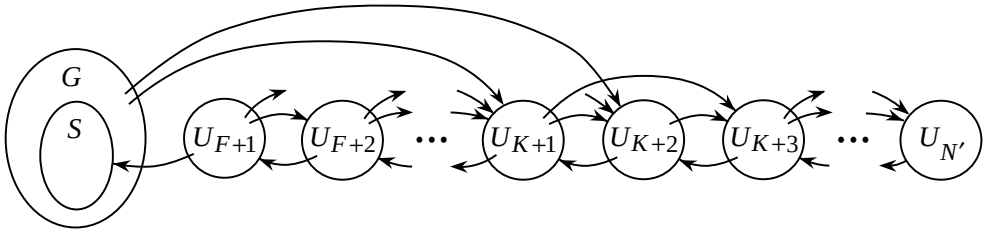

Figure 1: Structure of the CTMC models with the state cloning technique applied.

The class of models just described is quite large and encompasses, for instance, all the models of repairable systems which can be specified by the SAVE modeling language [15].

Let $X = \{X(t); t \geq 0\}$ be the finite irreducible CTMC modeling the system and let Ω be the state space of X. As in previous methods, bounds for the steady-state availability are computed using detailed knowledge of X in a subset G (the generated states) and bounding the behavior of X in $U = \Omega - G$. We will include in G all the states of the model with up to a given number K of failed components. We will also adopt the state cloning technique proposed in [23]. The technique can be explained as a modification of X in which clones of the states with a number of failed components $> F$ and $\leq K$ are added to U, accounting for the visits to the corresponding states of G after the number of failed components has been made greater than K and before it has fallen below $F + 1$. Let S be the "return" subset, i.e. the subset of G including the states through which the model can jump from U to G. We will consider in U the partition $\cup_{F+1 \leq k \leq N'} U_k$, $U_k = \{$states with k failed components$\}$, where $N' \leq N$, N being the number of components of the system modeled by X. The single component repair transition hypothesis implies the absence of transitions from U_k to U_l, $l < k - 1$ and from U_k to G, $k > F + 1$. Fig. 1 illustrates the structure of X.

Throughout the paper we will denote by λ_{ij}, $i, j \in \Omega$ the transition rate from state i to state j, by $\lambda_i = \sum_{j \in \Omega} \lambda_{ij}$, $i \in \Omega$ the output rate of state i, and by $\lambda_{iC} = \sum_{j \in C} \lambda_{ij}$, $i \in \Omega$, $C \subset \Omega$ the transition rate from i to subset C, all referred to X unless otherwise stated. We will also consider a number of transient CTMCs Y. Each such CTMC Y has a state space of the form $B \cup \{a\}$, where all states in B are transient and a is an absorbing state, and has a well-defined initial probability distribution with $P[Y(0) \in B] = 1$. We will denote by $\tau(i, Y)$, $i \in B$ the mean time spent by Y in i before absorption ($\tau(i, Y) = \int_0^\infty P[Y(t) = i]dt$). We will also use the notation $\tau(C, Y) = \sum_{i \in C} \tau(i, Y)$. It is well-known (see, for instance, [3]) that the mean times to absorption vector $\boldsymbol{\tau} = (\tau(i, Y))_{i \in B}$ is the solution of the linear system $\boldsymbol{\tau}^T \mathbf{A} = -\mathbf{q}^T$, where $\mathbf{A}$ is the restriction of the transition rate matrix of Y to B and $\mathbf{q} = (P[Y(0) = i])_{i \in B}$. The expected number of times that a transition (i, j) with rate λ_{ij} is followed is $\mu_{ij} = \tau(i, Y)\lambda_{ij}$. The result follows easily: $\mu_{ij} = \int_0^\infty P[Y(t) = i]\lambda_{ij}dt = \lambda_{ij} \int_0^\infty P[Y(t) = i]dt = \lambda_{ij}\tau(i, Y)$. It can be similarly shown that, given a partition $B \cup B^c$ of the state space of X and assuming $X(0) \in B$, the probability that X enters B^c through a state $j \in B^c$ is $\sum_{i \in B'} \tau(i, Y_B)\lambda_{ij}$, where Y_B is the transient CTMC tracking X till the exit of B (Y_B has state space $B' \cup \{a\}$, where a is an absorbing state and B' is the subset of B including all states reachable before exit from B from states with non-null initial probability, the same initial probability distribution and transition rates among states in B' as X, and transition

rates $\lambda'_{i,a} = \lambda_{i,B^c}$, so that Y_B enters a whenever X exits B). Note that $\tau(i, Y_B) > 0$, $i \in B'$.

Since the steady-state availability A is typically very close to 1, it is more convenient to consider the complementary steady-state unavailability measure $UA = 1 - A$. With D denoting the subset of down states of X and $\mathbf{p} = (p_i)_{i\in\Omega}$ the steady-state probability vector of X, UA is given by:

$$UA = \sum_{i\in D} p_i \,.$$

Let X_s, $s \in S$ be the CTMC obtained from X by redirecting to state s the transitions from states in U to S. Consider now the regenerative behavior of x_s with $X_s(0) = s$ (X_s may be in general non-irreducible) defined by the times at which X_s hits s from U. Let T_s and C_s be, respectively, the expected duration of a regenerative cycle and the expected down time between recurrences. Using semi-regenerative Markov process theory [10, Section 10.6] we have

Theorem 1. *There exist β_s, $s \in S$ with $\beta_s > 0$, $\sum_{s\in S} \beta_s = 1$ such that $UA = (\sum_{s\in S} \beta_s C_s)/(\sum_{s\in S} \beta_s T_s)$.*

Let UA_s be the steady-state unavailability computed from X_s with $X_s(0) = s$. Regenerative process theory gives $UA_s = C_s/T_s$. We have

Corollary 1. $\min_{s\in S}\{UA_s\} \le UA \le \max_{s\in S}\{UA_s\}$.

Proof. The result follows easily from Theorem 1 considering that $UA_s = C_s/T_s$. □

Denoting by $C_{G,s}$ and $C_{U,s}$ the contributions of, respectively, the states in G and the states in U to C_s, and by $T_{G,s}$ and $T_{U,s}$ the contributions to T_s, we have:

$$UA_s = \frac{C_{G,s} + C_{U,s}}{T_{G,s} + T_{U,s}} \,.$$

Assume that upper bounds $[T_{U,s}]_{\rm ub}$ and $[C_{U,s}]_{\rm ub}$ for, respectively, $T_{U,s}$ and $C_{U,s}$ are available. Let

$$[UA_s]_{\rm lb} = \frac{C_{G,s}}{T_{G,s} + [T_{U,s}]_{\rm ub}} \,, \tag{1}$$

$$[UA_s]_{\rm ub} = \frac{C_{G,s} + [C_{U,s}]_{\rm ub}}{T_{G,s} + [C_{U,s}]_{\rm ub}} \,. \tag{2}$$

Then, we have:

Theorem 2. *$[UA_s]_{\rm lb} \le UA_s \le [UA_s]_{\rm ub}$, where $[UA_s]_{\rm lb}$ is given by (1) and $[UA_s]_{\rm ub}$ is given by (2).*

Proof. The proof of the lower bound is easy:

$$UA_s = \frac{C_{G,s} + C_{U,s}}{T_{G,s} + T_{U,s}} \ge \frac{C_{G,s}}{T_{G,s} + T_{U,s}} \ge \frac{C_{G,s}}{T_{G,s} + [T_{U,s}]_{\rm ub}} \,.$$

To prove the upper bound, let $g(x) = (C_{G,s} + x)/(T_{G,s} + x)$. Its first derivative is $dg/dx = (T_{G,s} - C_{G,s})/(T_{G,s} + x)^2 > 0$ since $C_{G,s} < T_{G,s}$ (the state o is operational and, since there is a path to o from any state in G, o is visited with non-null probability). Then, using $C_{U,s} \le T_{U,s}$:

$$UA_s = \frac{C_{G,s} + C_{U,s}}{T_{G,s} + T_{U,s}} \le \frac{C_{G,s} + C_{U,s}}{T_{G,s} + C_{U,s}} = g(C_{U,s}) \le g([C_{U,s}]_{\rm ub}) = \frac{C_{G,s} + [C_{U,s}]_{\rm ub}}{T_{G,s} + [C_{U,s}]_{\rm ub}} \,. \quad \square$$

Corollary 1 and Theorem 2 give the following lower and upper bounds $[UA]_{\rm lb}$, $[UA]_{\rm ub}$ for *UA*:

$$[UA]_{\rm lb} = \min_{s\in S}[UA_s]_{\rm lb}\,, \tag{3}$$

$$[UA]_{\rm ub} = \max_{s\in S}[UA_s]_{\rm ub}\,. \tag{4}$$

$T_{G,s}$ and $C_{G,s}$ can be expressed in terms of the mean times to absorption vector of the transient CTMC Y_G^s with initial state s tracking X from s till exit from G:

$$T_{G,s} = \sum_{i\in G}\tau(i, Y_G^s)\,, \tag{5}$$

$$C_{G,s} = \sum_{i\in G\cap D}\tau(i, Y_G^s)\,. \tag{6}$$

The derivation of $[T_{U,s}]_{\rm ub}$ and $[C_{U,s}]_{\rm ub}$ is more elaborate and is described and justified in the next section. This section ends with a result for transient CTMCs which is closely related to a similar result for irreducible DTMCs given in [11]. The result is used in the next section to justify the bounds $[T_{U,s}]_{\rm ub}$ and $[C_{U,s}]_{\rm ub}$.

Theorem 3 (Exact aggregation for transient CTMCs)**.** *Let $Y = \{Y(t); t \geq 0\}$ be a transient CTMC with state space $B \cup \{a\}$, where all states in B are transient and a is an absorbing state, transition rates λ_{ij}, $i \in B$, $j \in B \cup \{a\}$, $i \neq j$, and initial probability distribution $P[Y(0) = i] = \pi_i$, $i \in B$, $\sum_{i\in B}\pi_i = 1$. Assume $\tau(i, Y) > 0$ for all $i \in B$. Let $B_1 \cup B_2 \cup \ldots \cup B_n$ be a partition of B. Then, there exists a transient CTMC $Y' = \{Y'(t); t \geq 0\}$ (the exact aggregation of Y) with state space $\{b_1, b_2 \ldots, b_n\} \cup \{a\}$, transition rates $\lambda'_{b_k,b_l} = \sum_{i\in B_k} w_i^k \lambda_{i,B_l}$, $1 \leq k, l \leq n$, $k \neq l$ and $\lambda'_{b_k,a} = \sum_{i\in B_k} w_i^k \lambda_{i,a}$, $1 \leq k \leq n$, with $w_i^k > 0$, $\sum_{i\in B_k} w_i^k = 1$, and initial probability distribution $P[Y'(0) = b_k] = \pi'_k = \sum_{i\in B_k}\pi_i$, such that $\tau(b_k, Y') = \tau(B_k, Y)$.*

Proof. See Appendix A. □

3 Bounds $[T_{U,s}]_{\rm ub}$ and $[C_{U,s}]_{\rm ub}$

3.1 Bounds $[T_{U,s}]_{\rm ub}$

Let

$$f_i = \sum_{e\in E_i}\lambda_{\rm ub}(e)\,. \tag{7}$$

$[T_{U,s}]_{\rm ub}$, $s \in S$ is obtained using "bounding" transient CTMCs Y^{u_k} with initial state u_k and the state transition diagram of Fig. 2(b), where for each state u_l and each $i \in FC$, $l + i \leq N$, there is a transition with rate f_i from u_l to u_{l+i}.

Let Y_U^i, $i \in U$ be the transient CTMC with initial state i tracking X from i till exit from U and let T_U^i be the mean time to absorption of Y_U^i. Let Y_G^s, $s \in S$ be the transient CTMC with initial state

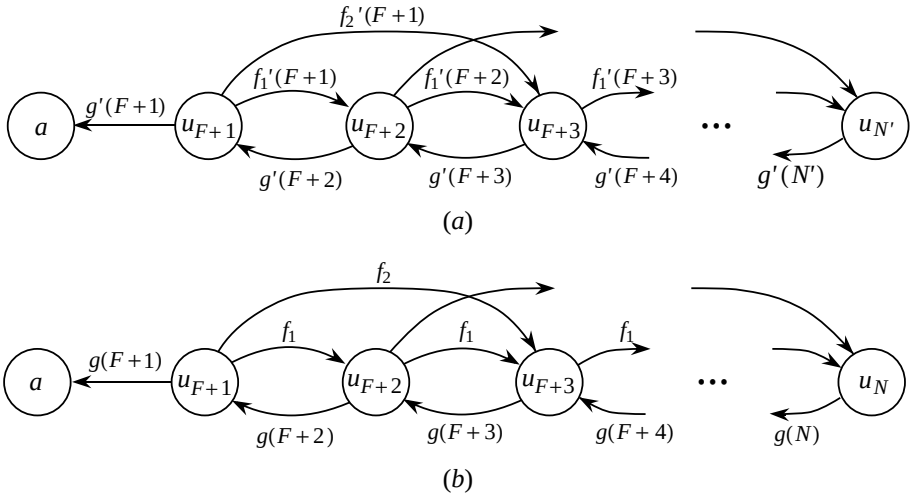

Figure 2: State transition diagrams of transient CTMCs $Y_U^{j'}$ (a) and Y^{u_k} (b).

s tracking X from s till exit from G. Noting that $\sum_{i\in G}\tau(i,Y_G^s)\lambda_{ij}$ is the probability that X with initial state $s\in S$ will enter U through state j, we have

$$T_{U,s}=\sum_{j\in U}\sum_{i\in G}\tau(i,Y_G^s)\lambda_{ij}T_U^j=\sum_{i\in G}\sum_{j\in U}\tau(i,Y_G^s)\lambda_{ij}T_U^j\,. \tag{8}$$

Let $T(k)$ be the mean time to absorption of Y^{u_k}. We will show that $T_{U,s}$ is upper bounded by

$$[T_{U,s}]_{\rm ub}=\sum_{k=F+1}^{N}\pi_k^sT(k)\,, \tag{9}$$

where

$$\pi_k^s=\sum_{i\in G}\tau(i,Y_G^s)\lambda_{i,U_k} \tag{10}$$

is the probability that X with initial state $s\in S$ will enter U through subset U_k. The proof requires the following lemma, closely related to the mean holding time lemma proved in [23]. Our version is required to prove Theorem 6.

Lemma 1. *Assume $N'\le N$. Let $Y'=\{Y'(t);t\ge 0\}$ be a transient CTMC with the state transition diagram of Fig. 2(a) and initial probability distribution $P[Y'(0)=u_i]=\pi_i$, $F+1\le i\le N'$, $\sum_{i=F+1}^{N'}\pi_i=1$. Let $Y=\{Y(t);t\ge 0\}$ be the transient CTMC with the state transition diagram of Fig. 2(b) and initial probability distribution $P[Y(0)=u_i]=\pi_i$, $F+1\le i\le N'$, $P[Y(0)=u_i]=0$, $N'<i\le N$. Assume $f_j\ge f_j'(i)$, $g(i)>0$, $F+1\le i\le N$, and $g(i)\le g'(i)$, $F+1\le i\le N'$. Then, $\tau(u_i,Y)\ge\tau(u_i,Y')$, $F+1\le i\le N'$.*

Proof. For notational conciseness, let $\tau_i=\tau(u_i,Y)$, $\tau_i'=\tau(u_i,Y')$. We will use as a basic tool the balance equation for a subset of states of a transient CTMC, which establishes that the initial probability of the subset plus the expected number of entries must be equal to the final probability of the subset plus the expected number of exits. The states u_i of Y and Y' are transient and, therefore, have final probabilities equal to 0.

The balance equation applied to Y' and the subset $\{u_{F+1}, u_{F+2}, \ldots, u_{N'}\}$ gives:

$$1 = \tau'_{F+1} g'(F+1)\,, \tag{11}$$

$$\tau'_{F+1} = \frac{1}{g'(F+1)}\,. \tag{12}$$

The balance equation applied to Y' and the subset $\{u_{F+1}, u_{F+2}, \ldots, u_{k-1}\}$, $F+1 < k \leq N'$ gives

$$\sum_{i=F+1}^{k-1} \pi_i + \tau'_k g'(k) = \tau'_{F+1} g'(F+1) + \sum_{i=F+1}^{k-1} \tau'_i \sum_{k-i \leq j \leq N'-i} f'_j(i)\,,$$

which, using (11) and $1 - \sum_{i=F+1}^{k-1} \pi_i = \sum_{i=k}^{N'} \pi_i$ gives

$$\tau'_k = \frac{\displaystyle\sum_{i=k}^{N'} \pi_i + \sum_{i=F+1}^{k-1} \tau'_i \sum_{k-i \leq j \leq N'-i} f'_j(i)}{g'(k)}\,. \tag{13}$$

Eqs. (12) and (13) define a recursive solution for τ'_k, $k = F+1, \ldots, N'$. Analysis of Y gives similar equations for τ_k (it has been used $P[Y(0) = u_i] = 0$ for $N' < i \leq N$):

$$\tau_{F+1} = \frac{1}{g(F+1)}, \tag{14}$$

$$\tau_k = \frac{\displaystyle\sum_{i=k}^{N} \pi_i + \sum_{i=F+1}^{k-1} \tau_i \sum_{k-i \leq j \leq N-i} f_j}{g(k)} = \frac{\displaystyle\sum_{i=k}^{N'} \pi_i + \sum_{i=F+1}^{k-1} \tau_i \sum_{k-i \leq j \leq N-i} f_j}{g(k)}\,. \tag{15}$$

The result is proved inductively for $k = F+1, \ldots, N'$. Since $g(F+1) \leq g'(F+1)$, using (14) and (12):

$$\tau_{F+1} = \frac{1}{g(F+1)} \geq \frac{1}{g'(F+1)} = \tau'_{F+1}\,.$$

Assume $\tau_l \geq \tau'_l$, $F+1 \leq l < k$. Using (15), $g(k) \leq g'(k)$, $N' \leq N$, $f_j \geq f'_j(i)$, and (13):

$$\tau_k = \frac{\displaystyle\sum_{i=k}^{N'} \pi_i + \sum_{i=F+1}^{k-1} \tau_i \sum_{k-i \leq j \leq N-i} f_j}{g(k)} \geq \frac{\displaystyle\sum_{i=k}^{N'} \pi_i + \sum_{i=F+1}^{k-1} \tau'_i \sum_{k-i \leq j \leq N'-i} f'_j(i)}{g(k)} = \tau'_k\,. \quad \square$$

Theorem 4. *$T_{U,s} \leq [T_{U,s}]_{\mathrm{ub}}$, where $[T_{U,s}]_{\mathrm{ub}}$ is given by (9) and (10).*

Proof. Let $l \in U_k$ and consider the exact aggregation $Y_U^{ll'}$ of Y_U^l for the partition $\cup_{F+1 \leq k \leq N'_l} U_k^l$, where U_k^l is the subset of U_k including the states reachable from l before exit from U and $F+1 \leq N'_l \leq N'$. The state transition diagram of $Y_U^{ll'}$ looks like the state transition diagram of Fig. 2(a) with N' replaced by N'_l. Using the notation of Fig. 2(a) for the transition rates of $Y_U^{ll'}$ and invoking the exact aggregation theorem for transient CTMCs, we have $f'_j(k) = \sum_{i \in U_k^l} w_i^k \lambda_{i,U_{k+j}}$; $g'(k) = \sum_{i \in U_k^l} w_i^k \lambda_{i,U_{k-1}}$, $F+1 < k \leq N'_l$; $g'(F+1) = \sum_{i \in U_{F+1}^l} w_i^{F+1} \lambda_{i,S}$; with $w_i^k > 0$, $\sum_{i \in U_k^l} w_i^k =$

1. Also, $\tau(U_i^l, Y_U^l) = \tau(u_i, Y_U^{l\prime})$. The transition rates $\lambda_{i,U_{k+j}}$, $i \in U_k^l$ result from contributions associated with failure bags involving j components and, therefore, are upper bounded by f_j. The transition rates $\lambda_{i,U_{k-1}}$, $i \in U_k^l$, $F+1 < k \leq N_l'$ are associated with repairs and, therefore, are lower bounded by $g(k)$. Similarly, $\lambda_{i,S}$, $i \in U_{F+1}^l$ is lower bounded by $g(F+1)$. Using $w_i^k > 0$, $\sum_{i \in U_k^l} w_i^k = 1$, we have $f_j'(k) = \sum_{i \in U_k^l} w_i^k \lambda_{i,U_{k+j}} \leq \sum_{i \in U_k^l} w_i^k f_j = f_j$, $g'(k) = \sum_{i \in U_k^l} w_i^k \lambda_{i,U_{k-1}} \geq \sum_{i \in U_k^l} w_i^k g(k) = g(k)$, $F+1 < k \leq N_l'$, and $g'(F+1) = \sum_{i \in U_{F+1}^l} w_i^k \lambda_{i,S} \geq \sum_{i \in U_{F+1}^l} w_i^k g(F+1) = g(F+1)$. Also, since $l \in U_k$, $P[Y_U^{l\prime}(0) = u_k] = 1$, $Y_U^{l\prime}$ and Y^{u_k} satisfy the conditions of Lemma 1. Then, using $\tau(U_i^l, Y_U^l) = \tau(u_i, Y_U^{l\prime})$ and the lemma:

$$T_U^l = \sum_{i=F+1}^{N_l'} \tau(U_i, Y_U^l) = \sum_{i=F+1}^{N_l'} \tau(u_i, Y_U^{l\prime}) \leq \sum_{i=F+1}^{N_l'} \tau(u_i, Y^{u_k}) \leq \sum_{i=F+1}^{N} \tau(u_i, Y^{u_k}) = T(k)\,. \tag{16}$$

Grouping in (8) the contributions of the states $j \in U$ according to the subsets U_k and using (16):

$$\begin{aligned} T_{U,s} &= \sum_{i \in G} \sum_{k=F+1}^{N} \sum_{l \in U_k} \tau(i, Y_G^s) \lambda_{il} T_U^l \\ &\leq \sum_{i \in G} \sum_{k=F+1}^{N} \sum_{l \in U_k} \tau(i, Y_G^s) \lambda_{il} T(k) = \sum_{i \in G} \sum_{k=F+1}^{N} \tau(i, Y_G^s) \lambda_{i,U_k} T(k) \\ &= \sum_{k=F+1}^{N} \sum_{i \in G} \tau(i, Y_G^s) \lambda_{i,U_k} T(k) = \sum_{k=F+1}^{N} \pi_k^s T(k)\,. \quad \square \end{aligned}$$

3.2 Bounds $[C_{U,s}]_{\mathrm{ub}}$

The bounds $[C_{U,s}]_{\mathrm{ub}}$ are computed using the failure distance concept. Let x be any state of X. Then, the failure distance from x, $d(x)$, is defined as the minimum number of components which have to fail in addition to those already failed in x to take the system down. Since a state y is down if and only if $m \subset F(y)$ for some $m \in \mathit{MC}$, $d(x)$ can be computed as

$$d(x) = \min_{m \in \mathit{MC}} |m - F(x)|\,, \tag{17}$$

where $|b|$ denotes the cardinality of bag b.

Let $L = \min_{m \in \mathit{MC}} |m|$. Let $U_{k,d}$ be the subset of U including the states with k failed components and failure distance d. The pairs (k,d) for which $U_{k,d}$ might be $\neq \emptyset$ are constrained by:

$$F+1 \leq k \leq N\,,$$

$$d_m(k) = \max\{0, L-k\} \leq d \leq \min\{L, N-k\} = d_M(k)\,.$$

The constraints $d \geq 0$ and $d \leq L$ are obvious. The constraint $d \geq L-k$ results from the fact that in any state x with k failed components, at most k components are failed in any minimal cut m, implying $|m - F(x)| \geq |m| - k \geq L - k$ and $d(x) \geq L - k$. The constraint $d \leq N - k$

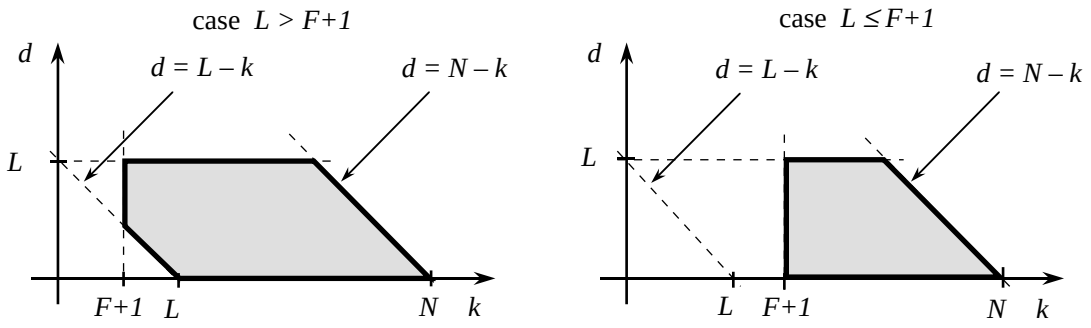

Figure 3: Possible shapes of $\mathcal{R}$.

results from the fact that in any state with k failed components there are $N-k$ unfailed components and, therefore, at most $N-k$ components are unfailed in any minimal cut. In the following we will call $\mathcal{R}$ the region of (k,d) pairs for which $U_{k,d}$ might be $\neq \emptyset$, as defined by the previous set of inequalities. $\mathcal{R}$ can have two shapes, illustrated in Fig. 3, depending on the values of F and L.

Let C_U^i be the mean down time to absorption of Y_U^i. Noting that $\sum_{i\in G}\tau(i,Y_G^s)\lambda_{ij}$ is the probability that X with initial state s will enter U through state j, $C_{U,s}$ can be expressed as

$$C_{U,s} = \sum_{j\in U}\sum_{i\in G}\tau(i,Y_G^s)\lambda_{ij}C_U^j = \sum_{i\in G}\sum_{j\in U}\tau(i,Y_G^s)\lambda_{ij}C_U^j\,. \tag{18}$$

Let $C(k,d)$ be upper bounds for C_U^i, $i\in U_{k,d}$. Let

$$\pi_{k,d}^s = \sum_{i\in G}\tau(i,Y_G^s)\lambda_{i,U_{k,d}} \tag{19}$$

be the probability that X with initial state $s\in S$ will enter U through subset $U_{k,d}$. Let

$$[C_{U,s}]_{\rm ub} = \sum_{k=F+1}^{N}\sum_{d=d_m(k)}^{d_M(k)}\pi_{k,d}^s C(k,d)\,. \tag{20}$$

We have:

Theorem 5. $C_{U,s}\le[C_{U,s}]_{\rm ub}$, *where* $[C_{U,s}]_{\rm ub}$ *is given by (20) and (19).*

Proof. Grouping in (18) the contributions of the states $j\in U$ according to the subsets $U_{k,d}$:

$$\begin{aligned}
C_{U,s} &= \sum_{i\in G}\sum_{k=F+1}^{N}\sum_{d=d_m(k)}^{d_M(k)}\sum_{j\in U_{k,d}}\tau(i,Y_G^s)\lambda_{ij}C_U^j\\
&\le \sum_{i\in G}\sum_{k=F+1}^{N}\sum_{d=d_m(k)}^{d_M(k)}\sum_{j\in U_{k,d}}\tau(i,Y_G^s)\lambda_{ij}C(k,d) = \sum_{i\in G}\sum_{k=F+1}^{N}\sum_{d=d_m(k)}^{d_M(k)}\tau(i,Y_G^s)\lambda_{i,U_{k,d}}C(k,d)\\
&= \sum_{k=F+1}^{N}\sum_{d=d_m(k)}^{d_M(k)}\sum_{i\in G}\tau(i,Y_G^s)\lambda_{i,U_{k,d}}C(k,d) = \sum_{k=F+1}^{N}\sum_{d=d_m(k)}^{d_M(k)}\pi_{k,d}^s C(k,d)\,. \quad \square
\end{aligned}$$

The bounds $C(k,d)$ are computed using an iterative procedure which starts with $C(k,d) = C(k)$, where $C(k)$ upper bounds C_U^i, $i \in U_k$, and improves the bounds using potentially better bounds $C'(k,d)$ until no further significant improvement is achieved.

The bounds $C(k)$ are

$$C(k) = \sum_{i=\max\{F+1,L\}}^{N} \tau(u_i, Y^{u_k}), \tag{21}$$

where Y^{u_k} is the transient CTMC with state transition diagram of Fig. 2(b) and initial state u_k.

Theorem 6. *For all $j \in U_k$, $C_U^j \leq C(k)$, where $C(k)$ is given by (21).*

Proof. The failure distance from a state with i failed components is $\geq L - i$. Then, $U_{i,0} = \emptyset$ for $i < L$. Let $Y_U^{j\prime}$ be the exact aggregation of Y_U^j for the partition $\cup_{F+1 \leq k \leq N_j'} U_k^j$, where U_k^j is the subset of U_k including the states reachable from j before exiting U and $F+1 \leq N_j' \leq N'$. Invoking the exact aggregation theorem for transient CTMCs and Lemma 1 as done in the proof of Theorem 4 we have $\tau(U_i^j, Y_U^j) = \tau(u_i, Y_U^{j\prime}) \leq \tau(u_i, Y^{u_k})$. Using also $N_j' \leq N$ and denoting by $U_{k,d}^j$ the subset of $U_{k,d}$ including the states reachable from j before exiting U:

$$\begin{aligned} C_U^j &= \sum_{i=F+1}^{N_j'} \sum_{l \in U_{i,0}^j} \tau(l, Y_U^j) = \sum_{i=\max\{F+1,L\}}^{N_j'} \sum_{l \in U_{i,0}^j} \tau(l, Y_U^j) \leq \sum_{i=\max\{F+1,L\}}^{N_j'} \sum_{l \in U_i^j} \tau(l, Y_U^j) \\ &= \sum_{i=\max\{F+1,L\}}^{N_j'} \tau(U_i^j, Y_U^j) = \sum_{i=\max\{F+1,L\}}^{N_j'} \tau(u_i, Y_U^{j\prime}) \leq \sum_{i=\max\{F+1,L\}}^{N} \tau(u_i, Y^{u_k}). \end{aligned}$$ □

Let $F(k,d,i,r)$, $(k,d,i,r) \in \mathcal{R}'$, where

$$\mathcal{R}' = \Big\{(k,d,i,r) : (k,d) \in \mathcal{R}, i \in FC, i \leq N-k, \max\{0, d-i\} \leq r \leq \min\{d, N-k-i\}\Big\},$$

be upper bounds for $\lambda_{l,\cup_{0 \leq d' \leq r} U_{k+i,d'}}$, $l \in U_{k,d}$, i.e. for the total failure rate involving i components from any state with k failed components and failure distance d to states with failure distance $\leq r$ (the constraints $i \leq N-k$ and $r \leq N-k-i$ result from imposing $(k+i, d') \in \mathcal{R}$ —see Fig. 3; the constraints $r \geq d-i$ and $r \leq d$ result from the fact that such a failure transition can neither increase the failure distance nor reduce it by more than i). The upper bounds $C'(k,d)$ are

$$\begin{aligned} C'(k,d) = {} & \frac{I(d=0)}{g(k)} \\ & + I(k > F+1) \Big[I(d > L-k)\, C(k-1,d) + I(d \leq L-k)\, C(k-1,d+1) \Big] \\ & + \frac{1}{g(k)} \sum_{\substack{i \in FC \\ i \leq N-k}} \sum_{j=\max\{0,k+d+i-N\}}^{\min\{i,d\}} f_{ij}(k,d)\, C(k+i, d-j), \end{aligned} \tag{22}$$

where $I(c)$ denotes the indicator function, returning 1 if c is true and 0 if c is false, and:

$$\begin{aligned} f_{i,j}(k,d) = F(k,d,i,d-j) - F(k,d,i,d-j-1), \\ \max\{0, k+d+i-N\} \leq j < \min\{i,d\}, \end{aligned} \tag{23}$$

Algorithm *Compute_Cbounds*($C(k)$, $f_{ij}(k,d)$, *tol*, $C(k,d)$)

for (all $(k,d) \in \mathcal{R}$) $C(k,d) = C(k)$;
do {
  max_rel_imp= 0;
  for ($k = F+1$; $k \leq N$; k++)
    for ($d = \max\{0, L-k\}$; $d \leq \min\{L, N-k\}$; d++){
      Compute $C'(k,d)$ using (22);
      if ($C'(k,d) < C(k,d)$){
        max_rel_imp $= \max\{$*max_rel_imp*$, (C(k,d) - C'(k,d))/C'(k,d)\}$;
        $C(k,d) = C'(k,d)$;
      }
    }
}
while (*max_rel_imp* $\geq$ *tol*);

Figure 4: Algorithm to compute $C(k,d)$ bounds.

$$f_{i,\min\{i,d\}}(k,d) = F(k,d,i,d-\min\{i,d\})\,. \tag{24}$$

The algorithm to compute the bounds $C(k,d)$ is given in Fig. 4. The parameter *tol* is a tolerance factor which determines when the improvement is sufficiently small for the iterative process to stop.

We prove next that the values $C(k,d)$ returned by the iterative improvement algorithm upper bound C_U^l, $l \in U_{k,d}$ if $F(k,d,i,r)$, $(k,d,i,r) \in \mathcal{R}'$ and $F(k,d,i,d)$, $(k,d,i,d) \in \mathcal{R}'$ are decreasing on d. The proof is done in a sequence of three propositions and a theorem, and depends on the fact that the bounds $C(k,d)$ are improved grouped by k.

Proposition 1. *Assume that $C(k,d)$, $(k,d) \in \mathcal{R}$ is decreasing on d. Then, for all $l \in U_{k,d}$, $C_U^l \leq C'(k,d)$.*

Proof. Let $l \in U_{k,d}$. C_U^l is equal to the mean time in l, if $d = 0$, plus the mean down time from the next visited state m, if $m \in U$. Since repair transitions involve just one component and, therefore, increase the failure distance by at most one:

$$\begin{aligned} C_U^l = {} & \frac{I(d=0)}{\lambda_l} \\ & + \; I(k > F+1)\left[I(d > L-k)\sum_{m\in U_{k-1,d}} \frac{\lambda_{lm}}{\lambda_l} C_U^m \; + \; I(d<L)\sum_{m\in U_{k-1,d+1}} \frac{\lambda_{lm}}{\lambda_l}\, C_U^m\right] \\ & + \sum_{\substack{i\in FC \\ i\leq N-k}} \sum_{j=\max\{0,k+d+i-N\}}^{\min\{i,d\}} \sum_{m\in U_{k+i,d-j}} \frac{\lambda_{lm}}{\lambda_l} C_U^m\,, \end{aligned}$$

where the factor $I(k > F+1)I(d > L-k)$ results from imposing $(k-1,d) \in \mathcal{R}$, the factor $I(k > F+1)I(d < L)$ results from imposing $(k-1, d+1) \in \mathcal{R}$, and the limits for the indices i and j in the last term guarantee $(k+i, d-j) \in \mathcal{R}$ (see Fig. 3). It is taken into account that: (1) a failure transition involving i components cannot increase the failure distance and reduces the failure distance by at most i, and (2) the failure distance from the destination state m is ≥ 0. Noting that

C_U^m, $m \in U_{k',d'}$ is upper bounded by $C(k',d')$ and introducing the notation $g_j(l) = \lambda_{l,U_{k-1,d+j}}$, $f_{ij}(l) = \lambda_{l,U_{k+i,d-j}}$, $J_m(i) = \max\{0, k+d+i-N\}$, and $J_M(i) = \min\{i,d\}$:

$$C_U^l \leq T_1 + T_2 + \sum_{\substack{i \in FC \\ i \leq N-k}} T_3(i)\,,$$

with

$$\begin{aligned}
T_1 &= \frac{I(d=0)}{\lambda_l}, \\
T_2 &= I(k > F+1)\Big[I(d > L-k)\,\frac{g_0(l)}{\lambda_l}\,C(k-1,d) \;+\; I(d<L)\,\frac{g_1(l)}{\lambda_l}\,C(k-1,d+1)\Big], \\
T_3(i) &= \sum_{j=J_m(i)}^{J_M(i)} \frac{f_{ij}(l)}{\lambda_l}\,C(k+i,d-j)\,.
\end{aligned}$$

Since $\lambda_l \geq g(k)$, we have:

$$T_1 \leq \frac{I(d=0)}{g(k)}\,.$$

To bound T_2, we consider first the case $k > F+1$, $d > L-k$. Since $C(k,d)$ is decreasing on d and $\lambda_l \geq g_0(l) + g_1(l)$:

$$\begin{aligned}
T_2 &= \frac{g_0(l)}{\lambda_l}\,C(k-1,d) \;+\; I(d<L)\,\frac{g_1(l)}{\lambda_l}\,C(k-l,d+1) \\
&\leq \left(\frac{g_0(l)}{\lambda_l} + I(d<L)\,\frac{g_1(l)}{\lambda_l}\right) C(k-1,d) \;\leq\; \frac{g_0(l)+g_1(l)}{\lambda_l}\,C(k-1,d) \;\leq\; C(k-1,d)\,.
\end{aligned}$$

For the case $k > F+1$, $d \leq L-k$ (which implies $d < L$), we have

$$T_2 = \frac{g_1(l)}{\lambda_l}\,C(k-1,d+1) \leq C(k-1,d+1)\,.$$

For $k \leq F+1$, $T_2 = 0$. Thus, in summary:

$$T_2 \leq I(k > F+1)\,\Big[I(d > L-k)\,C(k-1,d) \;+\; I(d \leq L-k)\,C(k-1,d+1)\Big].$$

To bound $T_3(i)$ we introduce the notation $F_{ir}(l) = \sum_{j=d-r}^{J_M(i)} f_{ij}(l)$, $d - J_M(i) \leq r \leq d - J_m(i)$. Note that $F_{ir}(l)$ is the sum of the failure transition rates from l involving i components and leading to states with failure distance $\leq r$. Thus, $F_{ir}(l) \leq F(k,d,i,r)$. Also, $f_{i,J_M(i)}(l) = F_{i,d-J_M(i)}(l)$ and $f_{ij}(l) = F_{i,d-j}(l) - F_{i,d-j-1}(l)$, $J_m(i) \leq j < J_M(i)$. Finally, $\lambda_l \geq g(k)$. Using all these relationships and (23) and (24):

$$\begin{aligned}
T_3(i) &= \sum_{j=J_m(i)}^{J_M(i)-1} \frac{F_{i,d-j}(l) - F_{i,d-j-1}(l)}{\lambda_l}\,C(k+i,d-j) \\
&\qquad + \;\frac{F_{i,d-J_M(i)}(l)}{\lambda_l}\,C(k+i,d-J_M(i)) \\
&= \frac{F_{i,d-J_m(i)}(l)}{\lambda_l}\,C(k+i,d-J_m(i))
\end{aligned}$$

$$
\begin{aligned}
&\quad + \sum_{j=J_m(i)+1}^{J_M(i)} \frac{F_{i,d-j}(l)}{\lambda_l} \Big(C(k+i,d-j) - C(k+i,d-j+1) \Big) \\
&\leq \frac{F(k,d,i,d-J_m(i))}{g(k)}\, C(k+i,d-J_m(i)) \\
&\quad + \sum_{j=J_m(i)+1}^{J_M(i)} \frac{F(k,d,i,d-j)}{g(k)} \Big(C(k+i,d-j) - C(k+i,d-j+1) \Big) \\
&= \sum_{j=J_m(i)}^{J_M(i)-1} \frac{F(k,d,i,d-j) - F(k,d,i,d-j-1)}{g(k)}\, C(k+i,d-j) \\
&\quad + \frac{F(k,d,i,d-J_M(i))}{g(k)}\, C(k+i,d-J_M(i)) \\
&= \sum_{j=J_m(i)}^{J_M(i)-1} \frac{f_{ij}(k,d)}{g(k)}\, C(k+i,d-j) \;+\; \frac{f_{i,J_M}(i)(k,d)}{g(k)}\, C(k+i,d-J_M(i)) \\
&= \frac{1}{g(k)} \sum_{j=J_m(i)}^{J_M(i)} f_{ij}(k,d)\, C(k+i,d-j)\,. \quad \square
\end{aligned}
$$

Proposition 2. *Assume that* $C(k,d)$, $(k,d) \in \mathcal{R}$, $F(k,d,i,r)$, $(k,d,i,r) \in \mathcal{R}'$, *and* $F(k,d,i,d)$, $(k,d,i,d) \in \mathcal{R}'$ *are decreasing on* d. *Then,*

$$
A(k,d,i) = \sum_{j=\max\{0,k+d+i-N\}}^{\min\{i,d\}} f_{ij}(k,d)\, C(k+i,d-j), \;\; i \in FC, i \leq N-k
$$

is decreasing on d.

Proof. Let $(k,d),(k,d+1) \in \mathcal{R}$. For notational conciseness let $j_m(d) = \max\{0, k+d+i-N\}$, $j_M(d) = \min\{i,d\}$. We start by obtaining an expression for $A(k,d,i)$ in terms of $C(k,d)$'s and $F(k,d,i,r)$'s. Using (23) and (24) and making the index change $r = d-j$:

$$
\begin{aligned}
A(k,d,i) &= \sum_{j=j_m(d)}^{j_M(d)} f_{ij}(k,d)\, C(k+i,d-j) \\
&= \sum_{j=j_m(d)}^{j_M(d)-1} \Big(F(k,d,i,d-j) - F(k,d,i,d-j-1) \Big)\, C(k+i,d-j) \\
&\quad + \; F(k,d,i,d-j_M(d))\, C(k+i,d-j_M(d)) \\
&= \sum_{r=d-j_M(d)+1}^{d-j_m(d)} \Big(F(k,d,i,r) - F(k,d,i,r-1) \Big)\, C(k+i,r) \\
&\quad + \; F(k,d,i,d-j_M(d))\, C(k+i,d-j_M(d)) \\
&= \sum_{r=d-j_M(d)}^{d-j_m(d)-1} F(k,d,i,r)\, \Big(C(k+i,r) - C(k+i,r+1) \Big) \\
&\quad + \; F(k,d,i,d-j_m(d))\, C(k+i,d-j_m(d))\,.
\end{aligned}
$$

Similarly, for $A(k,d+1,i)$:

$$\begin{aligned} A(k,d+1,i) &= \sum_{r=d-j_M(d+1)+1}^{d-j_m(d+1)} F(k,d+1,i,r)\,\Big(C(k+i,r)-C(k+i,r+1)\Big) \\ &\quad +\; F(k,d+1,i,d-j_m(d+1)+1)\,C(k+i,d-j_m(d+1)+1)\,. \end{aligned}$$

It is easy to verify that, for $d<i$, $d-j_M(d+1)+1 = d-j_M(d) = 0$, and, for $d\geq i$, $d-j_M(d+1)+1 = d-i+1$ and $d-j_M(d) = d-i$. Also, for $k+d+i > N-1$, $d-j_m(d+1) = d-j_m(d)-1 = N-k-i-1$, and, for $k+d+i\leq N-1$, $d-j_m(d+1)=d$ and $d-j_m(d)-1=d-1$. Then, subtracting the expressions for $A(k,d,i)$ and $A(k,d+1,i)$ and rearranging terms:

$$\begin{aligned} &A(k,d,i)-A(k,d+1,i) = \\ &\quad \sum_{r=d-j_M(d+1)+1}^{d-j_m(d)-1} \Big(F(k,d,i,r)-F(k,d+1,i,r)\Big)\,\Big(C(k+i,r)-C(k+i,r+1)\Big) \\ &\qquad +\; I(d\geq i)\,F(k,d,i,d-i)\,\Big(C(k+i,d-i)-C(k+i,d-i+1)\Big) \\ &\qquad +\; F(k,d,i,d-j_m(d))\,C(k+i,d-j_m(d)) \\ &\qquad -\; I(k+d+i\leq N-1)\,F(k,d+1,i,d)\,\Big(C(k+i,d)-C(k+i,d+1)\Big) \\ &\qquad -\; F(k,d+1,i,d-j_m(d+1)+1)\,C(k+i,d-j_m(d+1)+1)\,. \end{aligned}$$

The assumed monotonic properties for $F(k,d,i,r)$ and $C(k,d)$ ensure that the two first terms are ≥ 0. Then, it is enough to prove that the sum of the three last terms, which will be called $B(k,d,i)$, is ≥ 0. For $k+d+i > N-1$, $d-j_m(d) = d-j_m(d+1)+1 = N-k-i$ and

$$\begin{aligned} B(k,d,i) &= F(k,d,i,N-k-i)\,C(k+i,N-k-i) \\ &\quad -\; F(k,d+1,i,N-k-i)\,C(k+i,N-k-i) \\ &= \Big(F(k,d,i,N-k-i)-F(k,d+1,i,N-k-i)\Big)\,C(k+i,N-k-i) \;\geq\; 0\,. \end{aligned}$$

For $k+d+i\leq N-1$, $d-j_m(d)=d$, $d-j_m(d+1)+1=d+1$, and

$$\begin{aligned} B(k,d,i) &= F(k,d,i,d)\,C(k+i,d)-F(k,d+1,i,d)\,\Big(C(k+i,d)-C(k+i,d+1)\Big) \\ &\quad -\; F(k,d+1,i,d+1)\,C(k+i,d+1) \\ &= \Big(F(k,d,i,d)-F(k,d+1,i,d)\Big)\,\Big(C(k+i,d)-C(k+i,d+1)\Big) \\ &\quad +\; \Big(F(k,d,i,d)-F(k,d+1,i,d+1)\Big)\,C(k+i,d+1) \;\geq\; 0\,. \quad \square \end{aligned}$$

Proposition 3. *Assume that $C(k,d)$, $(k,d)\in\mathcal{R}$, $F(k,d,i,r)$, $(k,d,i,r)\in\mathcal{R}'$, and $F(k,d,i,d)$, $(k,d,i,d)\in\mathcal{R}'$ are decreasing on d. Then, $C'(k,d)$, $(k,d)\in\mathcal{R}$ is decreasing on d.*

Proof. Let $(k,d),(k,d+1)\in\mathcal{R}$. Using (22):

$$C'(k,d)-C'(k,d+1) = T_1+T_2+\sum_{\substack{i\in FC\\ i\leq N-k}} T_3(i)\,,$$

with:

$$\begin{aligned} T_1 &= \frac{I(d=0) - I(d+1=0)}{g(k)}, \\ T_2 &= I(k > F+1)\,\Big[I(d > L-k)\,C(k-1,d) \;+\; I(d \le L-k)\,C(k-1,d+1) \\ &\quad - I(d+1 > L-k)\,C(k-1,d+1) \;-\; I(d+1 \le L-k)\,C(k-1,d+2)\Big], \\ T_3(i) &= \frac{A(k,d,i) - A(k,d+1,i)}{g(k)}\,, \end{aligned}$$

where $A(k,d,i)$ is as defined in Proposition 2. $(k,d) \in \mathcal{R}$ implies $d \ge 0$, $d+1 > 0$, and $T_1 = I(d=0)/g(k) \ge 0$. To show $T_2 \ge 0$, we consider the cases: (1) $k = F+1$, (2) $k > F+1$, $d > L-k$, and (3) $k > F+1$, $d = L-k$. These cases cover all the possibilities since $(k,d) \in \mathcal{R}$ implies $k \ge F+1$ and $d \ge L-k$ (see Fig. 3). In case (1), $T_2 = 0$. In case (2), $T_2 = C(k-1,d) - C(k-1,d+1) \ge 0$. In case (3), $T_2 = C(k-1,d+1) - C(k-1,d+1) = 0$. $T_3(i)$ is ≥ 0 by Proposition 2. Then $C'(k,d) \ge C'(k,d+1)$. □

Theorem 7. *Assume that $F(k,d,i,r)$, $(k,d,i,r) \in \mathcal{R}'$ and $F(k,d,i,d)$, $(k,d,i,d) \in \mathcal{R}'$ are decreasing on d. Then, the bounds $C(k,d)$ obtained by algorithm Compute_Cbounds(k,d) (Fig. 4) are correct and decreasing on d.*

Proof. We consider the algorithm split into phases and prove inductively that the bounds $C^m(k,d)$, $m \ge 0$ available at the beginning ($m = 0$) and after each phase $m > 0$ are correct and decrease on d. Each phase includes the updating operations performed for a given value of k (loop of algorithm with control variable d). $C^0(k,d) = C(k)$, which are correct and decreasing (but, of course, not strictly) on d. Assume that the bounds available after phase m, $C^m(k,d)$, are correct and decreasing on d. Let k' be the value of k for which the bounds are updated in phase $m+1$. According to (22), $C^{m+1}(k',d)$ only depend on $C^m(k,d)$ for $k \ne k'$, and all $C^{m+1}(k',d)$ are computed using the same set of bounds $C^m(k,d)$. Then, Proposition 1 guarantees that $C'(k',d)$ are correct, and Proposition 3 that they are decreasing on d. Using the induction hypothesis, this implies that $C^{m+1}(k',d) = \min\{C^m(k',d), C'(k',d)\}$ are correct and decreasing on d. □

3.3 Bounds $F(k,d,i,r)$

Analysis of equations (22)–(24) reveals that the tighter the bounds $F(k,d,i,r)$ are, the tighter the final bounds $C(k,d)$ (and therefore $[C_{U,s}]_{\mathrm{ub}}$) will be. Also, for the procedure to compute the bounds $C(k,d)$ to be correct $F(k,d,i,r)$, $(k,d,i,r) \in \mathcal{R}'$ and $F(k,d,i,d)$, $(k,d,i,d) \in \mathcal{R}'$ must be decreasing on d. A set of bounds $F(k,d,i,r)$ satisfying these requirements which is relatively inexpensive to compute and is quite tight for small values of k can be derived using the concepts of importance and activity of failure bags. The *importance* $\mathrm{Imp}(e)$ of a failure bag e is defined as the minimum number of components which are left unfailed in any minimal cut affected by the failure bag. The *activity* $\mathrm{Act}(e)$ of a failure bag e is defined as the maximum number of components of the

failure bag in any minimal cut. Formally:

$$\text{Imp}(e) = \min_{\substack{m \in \mathit{MC} \\ m \cap e \neq \emptyset}} |m - e| \,, \tag{25}$$

$$\text{Act}(e) = \max_{m \in \mathit{MC}} |m \cap e| \,. \tag{26}$$

Consider a state x with k failed components and failure distance d and another state y reached from it through a failure bag e with cardinality i. Thus, we have (17) $d = \min_{m \in \mathit{MC}} |m - F(x)|$. Clearly, $d(y) \leq d$. Also, imposing $(k+i, d(y)) \in \mathcal{R}$, we have (see Fig. 3) $d(y) \leq N - k - i$. Therefore, we have $d(y) \leq \min\{d, N-k-i\}$. Lower bounds for $|m - F(y)|$, $m \in \mathit{MC}$ can be derived as follows considering $F(y) = F(x) + e$. First, for $m \cap e = \emptyset$:

$$|m - F(y)| = |m - F(x) - e| = |m - F(x)| \geq d \,,$$

and for $m \cap e \neq \emptyset$

$$|m - F(y)| = |m - F(x) - e| \geq |m - e| - |F(x)| = |m - e| - k \,.$$

Also, for any e:

$$|m - F(y)| = |m - F(x) - e| \geq |m - F(x)| - |m \cap e| \geq d - |m \cap e| \,.$$

Then, assumming $m \cap e \neq \emptyset$ for some $m \in \mathit{MC}$ for $d(y)$ we have

$$\begin{aligned} d(y) &= \min_{m \in \mathit{MC}} |m - F(y)| = \min\Big\{ \min_{\substack{m \in \mathit{MC} \\ m \cap e = \emptyset}} |m - F(y)|, \min_{\substack{m \in \mathit{MC} \\ m \cap e \neq \emptyset}} |m - F(y)| \Big\} \\ &\geq \min\Big\{ d, \min_{\substack{m \in \mathit{MC} \\ m \cap e \neq \emptyset}} |m - e| - k \Big\} = \min\{d, \text{Imp}(e) - k\} \,, \end{aligned}$$

and

$$d(y) = \min_{m \in \mathit{MC}} |m - F(y)| \geq \min_{m \in \mathit{MC}} \{d - |m \cap e|\} = d - \max_{m \in \mathit{MC}} |m \cap e| = d - \text{Act}(e) \,.$$

Since $d(y) \leq \min\{d, N-k-i\}$ for all failure bags $e \in E_i$, the transition rate to states with failure distance $\leq \min\{d, N-k-i\}$ due to failure bags with cardinality i is upper bounded by

$$F(k, d, i, \min\{d, N-k-i\}) = \sum_{e \in E_i} \lambda_{\text{ub}}(e) = f_i \,. \tag{27}$$

Consider now an r with $\max\{0, d-i\} \leq r < \min\{d, N-k-i\}$. If $m \cap e = \emptyset$ for all $m \in \mathit{MC}$, e cannot reduce the failure distance. Assume $m \cap e \neq \emptyset$ for some $m \in \mathit{MC}$, since $d(y)$ is lower-bounded by $\min\{d, \text{Imp}(e) - k\}$ and $d - \text{Act}(e)$, only failure bags e satisfying $\text{Imp}(e) - k \leq r$ and $d - \text{Act}(e) \leq r$ can lead to states with failure distance $\leq r$. The transition rate to such states due to failure bags with cardinality i is upper bounded by

$$F(k, d, i, r) = \sum_{\substack{e \in E_i \\ m \cap e \neq \emptyset \text{ for some } m \in \mathit{MC} \\ \text{Imp}(e) \leq k + r \\ \text{Act}(e) \geq d - r}} \lambda_{\text{ub}}(e), \quad \max\{0, d-i\} \leq r < \min\{d, N-k-i\} \,. \tag{28}$$

It is easy to check that the derived $F(k,d,i,r)$, $(k,d,i,r) \in \mathcal{R}'$ are decreasing on d. Also, for $(k,d,i,d) \in \mathcal{R}'$, $F(k,d,i,d) = \sum_{e \in E_i} \lambda_{\rm ub}(e)$, independent on d, and, therefore, $F(k,d,i,d)$ is (not strictly) decreasing on d. Thus, the derived $F(k,d,i,r)$ bounds satisfy the requirements imposed by Theorem 7.

4 Implementation details and algorithmic description

4.1 An algorithmic technique to compute the bounds solving five linear systems of size $|G|$

According to (1)–(4), $[UA]_{\rm lb}$ and $[UA]_{\rm ub}$ can be expressed as

$$[UA]_{\rm lb} = \min_{s \in S} \Big\{ \frac{C_{G,s}}{T_{G,s} + [T_{U,s}]_{\rm ub}} \Big\}, \tag{29}$$

$$[UA]_{\rm ub} = \max_{s \in S} \Big\{ \frac{C_{G,s} + [C_{U,s}]_{\rm ub}}{T_{G,s} + [C_{U,s}]_{\rm ub}} \Big\}. \tag{30}$$

Direct computation of $T_{G,s}$, $C_{G,s}$, $[T_{U,s}]_{\rm ub}$ and $[C_{U,s}]_{\rm ub}$, $s \in S$, using (5), (6), (9), (10), (19) and (20), involves the computation of $\tau(i, Y_G^s)$, $i \in G$, $s \in S$. This can be done by solving the $|S|$ linear systems:

$$\boldsymbol{\tau}_G^{sT} \mathbf{A}_G = -\mathbf{e}_s^T, \quad s \in S\,,$$

where $\boldsymbol{\tau}_G^s = (\tau(i, Y_G^s))_{i \in G}$, $\mathbf{A}_G$ is the restriction of the transition rate matrix of X to G, and $\mathbf{e}_s$ is a vector with component associated to state s equal to 1 and all other components equal to 0. Clearly, this procedure is very expensive when $|S|$ is large, as is the case when $F > 0$.

Consider the transient CTMC's Y_G^s with a reward rate structure v_i, $i \in G$ on them. The expected reward to absorption of Y_G^s, V_s, can be expressed as:

$$V_s = \sum_{i \in G} v_i \tau(i, Y_G^s)\,. \tag{31}$$

Let $C_s'' = C_{G,s} + [C_{U,s}]_{\rm ub}$, $T_s' = T_{G,s} + [T_{U,s}]_{\rm ub}$ and $T_s'' = T_{G,s} + [C_{U,s}]_{\rm ub}$. $[UA]_{\rm lb}$ and $[UA]_{\rm ub}$ can be expressed in terms of them and $C_{G,s}$ as (29), (30):

$$[UA]_{\rm lb} = \min_{s \in S} \Big\{ \frac{C_{G,s}}{T_s'} \Big\}, \tag{32}$$

$$[UA]_{\rm ub} = \max_{s \in S} \Big\{ \frac{C_s''}{T_s''} \Big\}. \tag{33}$$

Combining (5), (6), (9), (10), (19) and (20), we can write $C_{G,s}$, C_s'', T_s', and T_s'' in the form (31) with the reward rate structures v_i, $i \in G$:

$$v_i = I(i \in D) \quad \text{for} \quad C_{G,s}\,,$$

$$v_i = I(i \in D) + \sum_{k=F+1}^{N} \sum_{d=d_m(k)}^{d_M(k)} \lambda_{i,U_{k,d}} C(k,d) \quad \text{for} \quad C''_s \,,$$

$$v_i = 1 + \sum_{k=F+1}^{N} \lambda_{i,U_k} T(k) \quad \text{for} \quad T'_s \,,$$

$$v_i = 1 + \sum_{k=F+1}^{N} \sum_{d=d_m(k)}^{d_M(k)} \lambda_{i,U_{k,d}} C(k,d) \quad \text{for} \quad T''_s \,.$$

Then, we can adapt the second algorithm given in [5] to compute $C_{G,s}$, C'_s, T'_S, and T''_s, $s \in S$ solving only 5 linear systems, irrespectively of $|S|$. Without loss of generality assume that state 1 is the only state o in G without failed components, let $q_{ij} = \lambda_{ij}/\lambda_i$, let the matrix:

$$\tilde{\mathbf{B}} = \begin{pmatrix} 1 & -q_{12} & \cdots & -q_{1,|G|} \\ 0 & 1 & \cdots & -q_{2,|G|} \\ & & \cdots & \\ 0 & -q_{|G|,2} & \cdots & 1 \end{pmatrix},$$

and let the column vectors:

$$\tilde{\mathbf{C}} = (\tilde{C}_{G,i})_{i \in G} \,,$$

$$\tilde{\mathbf{C}}'' = (\tilde{C}''_i)_{i \in G} \,,$$

$$\tilde{\mathbf{T}}' = (\tilde{T}'_i)_{i \in G} \,,$$

$$\tilde{\mathbf{T}}'' = (\tilde{T}''_i)_{i \in G} \,,$$

$$\boldsymbol{\gamma} = (\gamma_i)_{i \in G} \,,$$

$$\mathbf{c} = (I(i \in D)/\lambda_i)_{i \in G} \,,$$

$$\mathbf{c}'' = (I(i \in D)/\lambda_i + \textstyle\sum_k \sum_d (\lambda_{i,U_{k,d}}/\lambda_i) C(k,d))_{i \in G} \,,$$

$$\boldsymbol{\mu}' = (1/\lambda_i + \textstyle\sum_k (\lambda_{i,U_k}/\lambda_i) T(k))_{i \in G} \,,$$

$$\boldsymbol{\mu}'' = (1/\lambda_i + \textstyle\sum_k \sum_d (\lambda_{i,U_{k,d}}/\lambda_i) C(k,d))_{i \in G} \,,$$

$$\boldsymbol{\omega} = (\lambda_{i,U}/\lambda_i)_{i \in G} \,.$$

The systems to be solved are:

$$\tilde{\mathbf{B}}\tilde{\mathbf{C}} = \mathbf{c} \,, \tag{34}$$

$$\tilde{\mathbf{B}}\tilde{\mathbf{C}}'' = \mathbf{c}'' \,, \tag{35}$$

$$\tilde{\mathbf{B}}\tilde{\mathbf{T}}' = \boldsymbol{\mu}' \,, \tag{36}$$

$$\tilde{\mathbf{B}}\tilde{\mathbf{T}}'' = \boldsymbol{\mu}'' \,, \tag{37}$$

$$\tilde{\mathbf{B}}\boldsymbol{\gamma} = \boldsymbol{\omega} \,. \tag{38}$$

$C_{G,s}$, C''_s, T'_s, T''_s, $s \in S$ can be computed from $\tilde{\mathbf{C}}$, $\tilde{\mathbf{C}}''$, $\tilde{\mathbf{T}}'$, $\tilde{\mathbf{T}}''$ and $\boldsymbol{\gamma}$ using:

$$C_{G,s} = \tilde{C}_{G,s} + \frac{1 - \gamma_s}{\gamma_1} \tilde{C}_{G,1} \,, \tag{39}$$

$$C''_s = \tilde{C}''_s + \frac{1-\gamma_s}{\gamma_1}\,\tilde{C}''_1\,, \tag{40}$$

$$T'_s = \tilde{T}'_s + \frac{1-\gamma_s}{\gamma_1}\,\tilde{T}'_1\,, \tag{41}$$

$$T''_s = \tilde{T}''_s + \frac{1-\gamma_s}{\gamma_1}\,\tilde{T}''_1\,. \tag{42}$$

The properties of the matrix $\tilde{\mathbf{B}}$ ensure the convergence of basic iterative methods (Gauss-Seidel and Jacobi) for the solution of the linear systems (34)–(38) (see [28] for the basic results and [5] for a detailed discussion). Experiments have shown that convergence under these methods is extremely fast [5]. Theoretical results explaining this fast convergence have also been obtained recently [17].

4.2 Computation of failure distances

In order to obtain the transition rates $\lambda_{i,U_{k,d}}$, $i \in G$ required by the method, it is necessary to compute the failure distances from the successors out of G of the states in the frontier of G. Since G includes all states with up to a given number of failed components, those successors will be reached through failure transitions. Let y be a successor of a state x in the frontier of G and let e be the failure bag causing the transition (x, y). The bag of failed components in y is $F(x) + e$ and both $F(x)$ and e are assumed to be known (see Section 3.1). Thus, we could compute $d(y)$ using (17) as

$$d(y) = \min_{m \in \mathit{MC}} |m - (F(x) + e)|\,. \tag{43}$$

However, such a procedure can be expensive if the number of minimal cuts is large. In this section we describe more sophisticated procedures which tend to be much less expensive when the number of minimal cuts is large.

We start by introducing the concept of *after minimal cut*. The after minimal cut associated with a minimal cut m and a failure bag $e \in E$ is $m' = m - e$. Let AMC_e be the set of after minimal cuts associated to failure bag e, i.e. $\mathit{AMC}_e = \{m' \mid m' = m - e, m \in \mathit{MC}, m \cap e \neq \emptyset\}$. Then, the failure distance from any state reached from x through a failure transition with failure bag e, $\mathrm{ad}(x, e)$, can be obtained as

$$\mathrm{ad}(x, e) = \min\{d(x), \min_{m \in \mathit{AMC}_e} |m - F(x)|\}\,. \tag{44}$$

Thus, we can obtain $\mathrm{ad}(x, e)$, $e \in E$ computing $d(x)$ by (17) and using (44). In this way the total number of minimal or after minimal cuts which are "touched" to compute $\mathrm{ad}(x, e)$, $e \in E$ is $|\mathit{MC}| + \sum_{e \in E} |\mathit{AMC}_e|$, which is typically much smaller than the number of minimal cuts which would be touched ($|E||\mathit{MC}|$) if the failure distances from all the successor states were computed using (43). Further reduction in the number of minimal cut "touches" and the associated overhead can be obtained by examining only minimal cuts or after minimal cuts which may reduce a known upper bound for, respectively, $d(x)$ or $\mathrm{ad}(x, e)$, $e \in E$. We assume that minimal cuts are indexed by their cardinality and selectors of up to a given cardinality R included in the minimal cut. The parameter R controls the degree of selection in the access to minimal cuts. Larger values of R

Algorithm *Compute_d*($F(x)$, L, $d(x)$)

$d(x) = L$;
for (increasing minimal cut cardinality c while $c < d(x) + |F(x)|$){
    $q = \min\{R, c - d(x) + 1\}$;
    for (each bag p of cardinality q included in $F(x)$)
        if (p is a selector of some minimal cut of cardinality c)
            for (each minimal cut m with $|m| = c$ and $p \subset m$)
                $d(x) = \min\{d(x), |m - F(x)|\}$;
}

Figure 5: Algorithm to compute failure distances.

yield fewer minimal cut "touches", but more potential selectors have to be tested. We have found $R = 2$ to be a good tradeoff in general. We assume the same indexing structure for the collection of after minimal cuts. We describe next two algorithms: the first one computes $d(x)$; the second one computes ad(x, e), $e \in E$, assuming $d(x)$ known.

The algorithm to compute $d(x)$ initializes the upper bound for $d(x)$, *ub*, to $L = \min_{m \in MC} |m|$. Since at most $|F(x)|$ components can be failed in any minimal cut we only need to consider the minimal cuts m with cardinality satisfying $|m| - |F(x)| < ub$, i.e. $|m| < ub + |F(x)|$. The minimal cuts to be considered can be further restricted by noting that $|m - F(x)|$ cannot be $< ub$ unless m contains a selector $p \subset F(x)$ and $|m| - |p| < ub$, i.e. $|p| \geq |m| - ub + 1$. Thus, for each possible minimal cut cardinality c, we can restrict our attention to the minimal cuts of cardinality c containing selectors $p \subset F(x)$ and $|p| = \min\{R, c - ub + 1\} = q$. Possible selectors $p \subset F(x)$ can be obtained by generating all bags of cardinality q included in $F(x)$. Then, if the selectors are kept in a hash table or a similarly efficient structure, it is possible to test whether each possible selector p is in fact a selector and, with the appropriate data structures, visit all minimal cuts of cardinality c including p. The discussion justifies the algorithm given in Fig. 5.

Assuming $d(x)$ known, similar ideas can be used to reduce the number of after minimal cuts which have to be examined to obtain ad(x, e), $e \in E$. To reduce the overhead associated with the control of the algorithm, only an upper bound *adub* for all ad(x, e), $e \in E$ is used. The after-failure distances ad(x, e) are initialized to $\min\{d(x), L_e\}$, where $L_e = \min_{m \in AMC_e} |m|$. The upper bound *adub* can be initialized to the maximum of the initial after-failure distances. $d(x)$ and L_e, $e \in E$ are passed to the algorithm. The algorithm is given in Fig. 6.

4.3 Computation of $T(k)$ and $C(k)$

$T(k)$ and $C(k)$ can be computed from the mean time to absorption vector of Y^{u_k} using (16) and (21), respectively. Letting $\boldsymbol{\tau}_U^{u_k} = (\tau(u_i, Y^{u_k}))_{F+1 \leq i \leq N}$ and denoting by $\mathbf{A}_U$ the restriction of the transition rate matrix of the CTMCs Y^{u_k} to its transient states, $\boldsymbol{\tau}_U^{u_k}$ can be obtained solving

$$\boldsymbol{\tau}_U^{u_k T} \mathbf{A}_U = -\mathbf{e}_{u_k}^T , \tag{45}$$

Algorithm *Compute_all_ad*($F(x)$, $d(x)$, L_e, ad(x,e))

for (each $e \in E$) ad$(x,e) = \min\{d(x), L_e\}$;
adub $= \max_{e\in E}\{\text{ad}(x,e)\}$;
for (increasing after minimal cut cardinality c while $c <$ *adub* $+ |F(x)|$){
    $q = \min\{R, \max\{1, c -$ *adub* $+ 1\}\}$;
    for (each bag p of cardinality q included in $F(x)$)
        if (p is a selector of some after minimal cut of cardinality c)
            for (each after minimal cut m with $|m| = c$ and $p \subset m$){
                Let e be the failure bag associated to m;
                ad$(x,e) = \min\{\text{ad}(x,e), |m - F(x)|\}$;
            }
}

Figure 6: Algorithm to compute after failure distances.

where $\mathbf{e}_{u_k}$ is a vector with the component associated with state u_k equal to 1 and all other components equal to 0. Thus, in principle, $T(k)$, $C(k)$, $F+1 \le k \le N$ could be computed solving $N-F$ linear systems (45). A more efficient procedure can however be developed as follows. Let

$$\lambda(k) = g(k) + \sum_{\substack{i\in FC \\ k+i\le N}} f_i\,, \tag{46}$$

be the output rate from state u_k of the state transition diagram of Fig. 2(b). Noting that $T(k)$ is equal to the mean time in u_k plus the mean time to absorption of $Y^{u_{k'}}$, with $u_{k'}$ being the next transient state visited, we obtain the equations:

$$T(k) = \frac{1}{\lambda(k)} + \frac{g(k)}{\lambda(k)}T(k-1) + \sum_{\substack{i\in FC \\ k+i\le N}} \frac{f_i}{\lambda(k)}T(k+i), \;\; F+1 < k < N\,, \tag{47}$$

$$T(N) = \frac{1}{g(N)} + T(N-1). \tag{48}$$

According to (21) $C(k)$ is the mean time in u_k if $k \ge L$ plus $C(k')$, with $u_{k'}$ being the next transient state visited. Then, using $L \le N$:

$$C(k) = \frac{I(k\ge L)}{\lambda(k)} + \frac{g(k)}{\lambda(k)}C(k-1) + \sum_{\substack{i\in FC \\ k+i\le N}} \frac{f_i}{\lambda(k)}C(k+i), \;\; F+1 < k < N\,, \tag{49}$$

$$C(N) = \frac{1}{g(N)} + C(N-1)\,. \tag{50}$$

Using (47) and (48) it is possible to solve recursively $T(k)$, $F+1 \le k < N$, in $T(N)$, yielding:

$$T(N-1) = T(N) - \frac{1}{g(N)}\,, \tag{51}$$

$$T(k) = \frac{1}{g(k+1)}\Big[\lambda(k+1)T(k+1) - 1 - \sum_{\substack{i\in FC \\ k+i+1\le N}} f_i T(k+i+1)\Big], \quad k = N-2,\ldots,F+1\,. \tag{52}$$

Similarly, using (49) and (50) it is possible to solve recursively $C(k)$, $F+1 \leq k < N$, in $C(N)$, yielding:

$$C(N-1) = C(N) - \frac{1}{g(N)}\,, \tag{53}$$

$$\begin{aligned} C(k) &= \frac{1}{g(k+1)}\Big[\lambda(k+1)C(k+1) - I(k+1 \geq L) - \sum_{\substack{i \in FC \\ k+i+1 \leq N}} f_i C(k+i+1)\Big] \\ & \qquad k = N-2, \ldots, F+1\,. \end{aligned} \tag{54}$$

Then, it is enough to solve (45) for $k = N$, use (16) and (21) to compute, respectively, $T(N)$ and $C(N)$, use (51) and (52) to compute the remaining $T(k)$'s, and use (53) and (54) to compute the remaining $C(k)$'s.

(45) for $k = N$ can be solved efficiently with direct methods exploiting the fact that, under the state ordering $u_{F+1}, \ldots, u_N$, $\mathbf{A}_U$ has an upper Hessenberg structure and all components except the last one of $\mathbf{e}_{u_N}$ are 0 (the last component is 1). Defining $\nu_i = \tau(u_i, Y^{u_N})/\tau(u_{F+1}, Y^{u_N})$, the first $N-F-1$ equations (i.e. all except the last one) give a triangular system on ν_i, $F+2 \leq i \leq N$ which can be easily solved. Substituting then $\nu_i \tau(u_{F+1}, Y^{u_N})$, $F+1 \leq i \leq N$, for $\tau(u_i, Y^{u_N})$ in the last equation and using the solution for ν_i, $F+2 \leq i \leq N$ found in the previous step gives an equation on $\tau(u_{F+1}, Y^{u_N})$. Solving that equation and using $\tau(u_i, Y^{u_N}) = \nu_i \tau(u_1, Y^{u_N})$, $F+2 \leq i \leq N$ we can obtain $\tau(u_i, Y^{u_N})$, $F+2 \leq i \leq N$. The solution procedure can be described as follows:

$$\nu_{F+1} = 1, \tag{55}$$

$$\nu_i = \frac{1}{g(i)}\Big[\lambda(i-1)\nu_{i-1} - \sum_{\substack{F+1 \leq j \leq i-2 \\ i-j-1 \in FC}} f_{i-j-1}\nu_j\Big],\ \ i = F+2, \ldots, N\,, \tag{56}$$

$$\tau(u_{F+1}, Y^{u_N}) = \frac{1}{\lambda(N)\nu_N - \sum_{\substack{F+1 \leq i \leq N-1 \\ N-i \in FC}} f_{N-i}\nu_i}\,, \tag{57}$$

$$\tau(u_i, Y^{u_N}) = \nu_i \tau(u_{F+1}, Y^{u_k})\,,\ \ i = F+2, \ldots, N\,. \tag{58}$$

4.4 Algorithmic description

For the sake of clarity a summary algorithmic description of the bounding method is given in Fig. 7. We make reference to the key equations obtained so far and the algorithms which improve iteratively the bounds $C(k,d)$ and compute the distance and after-failure distances given in Figures. 4–6. It should be noted that, since all states up to a given number K of failed components are generated, all transitions exiting the generated subset G are of the failure type. Thus, the distances to the successors out of G of states in G (required to compute $\alpha(i)$, $i \in G$ and $\beta(i)$, $i \in G$) can be obtained by combining the procedures *Compute_d*() and *Compute_all_ad*() described in Section 4.1 as done in the algorithm. Also, we do not store explicitly the matrix $\tilde{\mathbf{B}}$, since the elements of that matrix can be computed with little effort from the transition rates λ_{ij} and the output rates λ_i of X restricted to G.

Algorithm *Bound*(K, F, *tol*, $[UA]_{\rm lb}$, $[UA]_{\rm ub}$)

$L = \min_{m \in MC} |m|$;
for ($e \in E$) $L_e = \min_{m \in AMC_e} |m|$;
Compute f_i, $i \in FC$ using (7);
Compute $\tau(u_i, Y^{u_N})$, $F+1 \le i \le N$ using (55)–/(58);
Compute $T(N)$ using (16);
Compute $\lambda(k)$, $F+1 \le k < N$ using (46);
Compute $T(k)$, $k = N-1, \ldots, F+1$ using (51), (52);
Compute $C(N)$ using (21);
Compute $C(k)$, $k = N-1, \ldots, F+1$ using (53), (54);
Compute Imp(e), Act(e), $e \in E$ using (25), (26);
Compute $F(k,d,i,r)$, $F+1 \le k \le N$, $\max\{0, L-k\} \le d \le \min\{L, N-k\}$, $i \in FC$, $i \le N-k$, $\max\{0, d-i\} \le r \le \min\{d, N-k-i\}$ using (27), (28);
Compute $f_{ij}(k,d)$, $F+1 \le k \le N$, $\max\{0, L-k\} \le d \le \min\{L, N-k\}$, $i \in FC$, $i \le N-k$, $\max\{0, k+d+i-N\} \le j \le \min\{i, d\}$ using (23), (24);
Compute_Cbounds($C(k)$, $f_{ij}(k,d)$, *tol*, $C(k,d)$);
Generate the state transition diagram of X restricted to G (states with up to K failed components) starting from the state without failed components (state 1);
for ($i \in G$){
    if ($i \in D$) $c_i = c_i'' = 1/\lambda_i$;
    else $c_i = c_i'' = 0$;
    $\mu_i' = \mu_i'' = 1/\lambda_i$;
    $\omega_i = 0$;
    Let *Succ*(i) be the set of successors of i in X not included in G;
    if (*Succ*$(i) \ne \emptyset$){
        Compute_d($F(i)$, L, $d(i)$);
        Compute_all_ad($F(i)$, $d(i)$, L_e, ad(i,e));
        for ($j \in$ *Succ*(i)){
            Let e be the failure bag associated to transition (i,j);
            $k = |F(i)| + |e|$, $d =$ ad(i,e);
            c_i'' += $(\lambda_{ij}/\lambda_i) C(k,d)$;
            μ_i' += $(\lambda_{ij}/\lambda_i) T(k)$;
            μ_i'' += $(\lambda_{ij}/\lambda_i) C(k,d)$;
            ω_i += λ_{ij}/λ_i;
        }
    }
}
Let $\mathbf{c} = (c_i)_{i \in G}$, $\mathbf{c''} = (c_i'')_{i \in G}$, $\boldsymbol{\mu'} = (\mu_i')_{i \in G}$, $\boldsymbol{\mu''} = (\mu_i'')_{i \in G}$, $\boldsymbol{\omega} = (\omega_i)_{i \in G}$;
Solve by Gauss-Seidel the linear systems (34)–(38);
Let S be the subset of G including the states with F failed components;
Compute $C_{G,s}$, C_s'', T_s', and T_s'', $s \in S$ using (39)–(42);
Compute $[UA]_{\rm lb}$ and $[UA]_{\rm ub}$ using (32), (33);

Figure 7: Bounding algorithm.

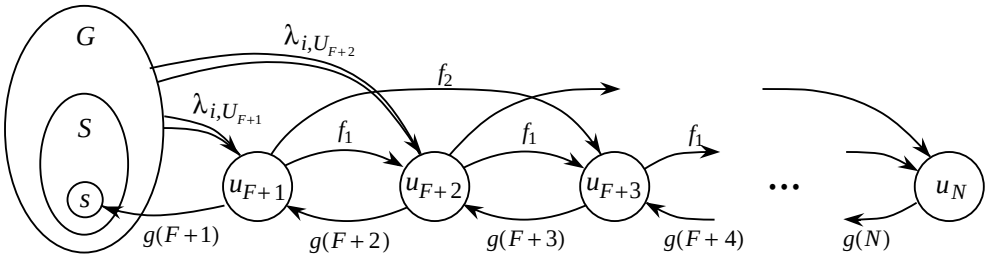

Figure 8: State transition diagrams of the CTMCs X_s used in the bounding method proposed in [22].

5 Comparison

In this section we compare the tightness of the bounds and the computational requirements of the method described here with those of the second algorithm given in [5], which obtains the same bounds as in [23] at a lower computational cost, especially for $F > 0$. We start by comparing the tightness of the bounds.

In the method proposed in [23], bounds for *UA* are obtained using CTMCs X'_s, $s \in S$ with the state transition diagram shown in Fig. 8. The portion of X'_s bounding the behavior in U (states $u_{F+1}, \ldots, u_N$) is identical to the state transition diagram of the transient CTMCs Y^{u_k} considered in our method (Fig. 2(b)). Each X'_s is associated to a possible return state $s \in S$. The bounds obtained in [23] are:

$$[UA]'_{\rm lb} = \min_{s \in S}\{[UA_s]'_{\rm lb}\}\,, \tag{59}$$

$$[UA]'_{\rm ub} = \max_{s \in S}\{[UA_s]'_{\rm ub}\}\,, \tag{60}$$

where $[UA_s]'_{\rm lb}$ is obtained from the steady-state solution of X'_s by assuming operational the states u_k, $F+1 \leq k \leq N$; and $[UA_s]'_{\rm ub}$ is obtained from the steady-state solution of X'_s by assuming down the states u_k, $F+1 \leq k \leq N$, (for both $[UA_s]'_{\rm lb}$ and $[UA_s]'_{\rm ub}$, states in G are taken operational or down as they truly are).

Consider the regenerative behavior of X'_s defined by the times at which X'_s enters s from u_{F+1}. Notice that the bounding portion of X'_s is entered once in each regenerative cycle and that the detailed portion (subset G) is entered through s. Given the relationships between X'_s and Y^s_G, Y^{u_k}, it is clear that the mean time spent by X'_s in the subset $\{u_{F+1}, \ldots, u_N\}$ in each regenerative cycle is $[T_{U,s}]_{\rm ub}$. Then, using regenerative process theory:

$$[UA_s]'_{\rm lb} = \frac{C_{G,s}}{T_{G,s} + [T_{U,s}]_{\rm ub}}\,, \tag{61}$$

$$[UA_s]'_{\rm ub} = \frac{C_{G,s} + [T_{U,s}]_{\rm ub}}{T_{G,s} + [T_{U,s}]_{\rm ub}}\,. \tag{62}$$

Comparing (61) with (1) we have $[UA_s]_{\rm lb} = [UA_s]'_{\rm lb}$ and, therefore (59), (3), $[UA]_{\rm lb} = [UA]'_{\rm lb}$. Regarding $[UA]_{\rm ub}$ and $[UA]'_{\rm ub}$ we have:

Theorem 8. $[UA]_{\rm ub} \leq [UA]'_{\rm ub}$ *and a sufficient condition for* $[UA]_{\rm ub} < [UA]'_{\rm ub}$ *is* $F + 1 < L$.

Proof. The algorithm *Compute_Cbounds*() returns bounds $C(k,d)$ which are $\leq C(k)$. Also, comparing (10) and (19) we have $\pi_k^s = \sum_{d=d_m(k)}^{d_M(k)} \pi_{k,d}^s$, with $d_m(k) = \max\{0, L-k\}$ and $d_M(k) = \min\{L, N-k\}$. Then, using (20):

$$[C_{U,s}]_{\rm ub} = \sum_{k=F+1}^{N} \sum_{d=d_m(k)}^{d_M(k)} \pi_{k,d}^s C(k,d) \leq \sum_{k=F+1}^{N} \sum_{d=d_m(k)}^{d_M(k)} \pi_{k,d}^s C(k) = \sum_{k=F+1}^{N} \pi_k^s C(k)\,.$$

Comparing (16) and (21) we see that $C(k) \leq T(k)$ and $C(k) < T(k)$ if $F+1 < L$. Then, we have (9):

$$[C_{U,s}]_{\rm ub} \leq \sum_{k=F+1}^{N} \pi_k^s T(k) = [T_{U,s}]_{\rm ub}\,,$$

with strict inequality guaranteed if $F+1 < L$. Then using (2) and (62):

$$[UA_s]_{\rm ub} = \frac{C_{G,s} + [C_{U,s}]_{\rm ub}}{T_{G,s} + [T_{U,s}]_{\rm ub}} \leq \frac{C_{G,s} + [T_{U,s}]_{\rm ub}}{T_{G,s} + [T_{U,s}]_{\rm ub}} = [UA_s]'_{\rm ub}\,,$$

with strict inequality guaranteed if $[C_{U,s}]_{\rm ub} < [T_{U,s}]_{\rm ub}$, i.e., if $F+1 < L$. The result follows considering (4) and (60). □

Thus, the bounds given by the method proposed here are never worse than the bounds given by the method proposed in [23] and are guaranteed to be better when $F+1 < L$. In practice, the bounds $C(k,d)$ tend to be much smaller than $C(k)$ for $d > 0$ and $[UA]_{\rm ub}$ tends to be much closer to UA than $[UA]'_{\rm ub}$ when U contains in its frontier a significant portion of operational states. In these circumstances, $[UA]'_{\rm ub}$ tends to be much looser than $[UA]'_{\rm lb}$, and the interval for UA defined by the bounds obtained here tends to be significantly smaller than the interval defined by the bounds obtained in [23]. This will be confirmed in the next section by numerical experiments.

The computational requirements of the method proposed here and the second algorithm proposed in [5] are similar[2]. For models for which tight bounds can be obtained with a reasonable number of detailed states (of the order of tens of thousands), the value of N (number of components of the system) is moderate (up to 100). Also values of L (minimum number of components which have to fail for the system to go down) larger than 3 are rare. Then, the storage of the structures required to obtain the bounds $C(k,d)$ (ignoring minimal cuts) is small. Further, this storage could be freed once the bounds $T(k)$ and $C(k,d)$ have been computed, except, of course, for the $T(k)$'s and $C(k,d)$'s corresponding to values of k of the states which may be in the frontier of G, and the number of such values is $|FC|$ in the worst case. Regarding the rest of the algorithm, the method proposed here needs the storage of six vectors of size $|G|$: $\mathbf{c}$, $\mathbf{c}''$, $\boldsymbol{\mu}'$, $\boldsymbol{\mu}''$ and $\boldsymbol{\omega}$ and the Gauss-Seidel iteration vector (the respective solution vectors $\tilde{\mathbf{C}}$, $\tilde{\mathbf{C}}''$, $\tilde{\mathbf{T}}'$, $\tilde{\mathbf{T}}''$ and $\boldsymbol{\gamma}$ can rewrite the right-hand side vectors once they are computed, $C_{G,s}$, $s \in S$ can rewrite $\tilde{\mathbf{C}}$, and so on), whereas the other method requires the storage of five vectors of size $|G|$. This storage overhead is small, especially since most of the storage is used for the restriction of X to G and the state descriptions required during the generation process.

[2]The first algorithm given in [5] requires the solution of $|W|+2$ linear systems of size $|G|$, where W is the set of indices k for which G has transitions to U_k. Thus, at best the first algorithm will require the solution of one less linear system than the second one.

The algorithms given in Section 4.1 to compute the failure distances require the knowledge of all minimal cuts of the structure function of the system. This adds three overheads: (1) time required to find the minimal cuts, (2) storage of minimal cuts, after minimal cuts, and corresponding access structures, and (3) time spent in the calls to the functions *Compute_d*() and *Compute_all_ad*(). The techniques incorporated in functions *Compute_d*() and *Compute_all_ad*() to reduce the number of cut "touches" are very efficient and make overhead (3) negligible when compared with overheads (1) and (2). These overheads could be significant when the structure function of the system has many minimal cuts (i.e., of the order of tens of thousands). An approach which is routinely used in fault-tree analysis (see, for instance, [24]) to limit the computational requirements is to generate only minimal cuts of cardinality $\leq M$. This approach can be used to reduce overheads (1) and (2) when $|MC|$ is very large. With this partial knowledge, we can assume pessimistically for the computation of the failure distances that the system is failed for all combinations of more than M failed components and obtain a less tight bound $[UA]_{\rm ub}$, but still $\leq [UA]'_{\rm ub}$. The modifications to the bounding algorithm given in Fig. 7 are simple: L and L_e should be computed by

$$L = \min\{M+1, \min_{m \in MC} |m|\}\,,$$

$$L_e = \min\Big\{\max\{0, M+1-|e|\}, \min_{m \in AMC_e} |m|\Big\},$$

and the importance and activity of failure bags should be computed as:

$$\mathrm{Imp}(e) = \min\Big\{\min_{\substack{m \in MC \\ m \cap e \neq \emptyset}} |m-e|, \max\{0, M+1-|e|\}\Big\},$$

$$\mathrm{Act}(e) = \max\Big\{\max_{m \in MC} |m \cap e|, \min\{M+1, |e|\}\Big\}.$$

The algorithm *Compute_d*() should be modified changing the initial $d(x)$ to

$$d(x) = \min\{L, \max\{0, M+1-|F(x)|\}\}\,.$$

The algorithm *Compute_all_ad*() should be modified changing the initial $\mathrm{ad}(x,e)$ to

$$\mathrm{ad}(x,e) = \min\{d(x), \min\{L_e, \max\{0, M+1-|F(x)|-|e|\}\}\}\,.$$

Taking $M = 0$ reduces the upper bound $[UA]_{\rm ub}$ to the upper bound $[UA]'_{\rm ub}$ obtained in [23] and no minimal cut has to be generated. Greater values of M give more minimal cuts and tighter upper bounds $[UA]_{\rm ub}$. Thus, we can trade off overheads associated to failure distance computations with bound tightness. As stated above, the tradeoff is only meaningful when the number of minimal cuts is large (of the order of tens of thousands). In the next section we will illustrate the dependence of the quality of the bounds on M.

6 Numerical Analysis

In this section we illustrate the bounding method and compare it in terms of bounds tightness with the method proposed in [23] using large examples. The examples are of fault-tolerant systems

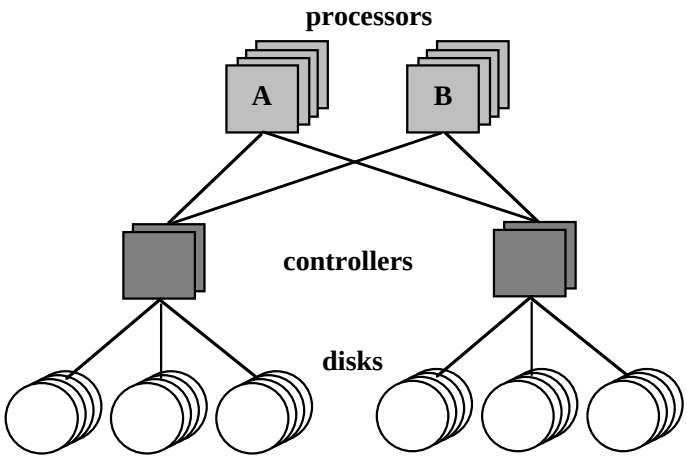

Figure 9: Block diagram for the first example.

with about 40 components and yielding state spaces with a number of states of the order of 10^{10}. We start considering the large example of [23], a distributed fault-tolerant database system, whose block diagram is given in Fig. 9. The system includes two processor types (A and B), two sets of dual-ported controllers with two controllers per set and six disk clusters of four disks. Each set of controllers controls three clusters. Each processor type has three spares. The system is operational if at least one processor of any type is unfailed, at least one controller in each set is unfailed and at least three disks in each cluster are unfailed. A failure in the active processor A is propagated to the active processor B with probability 0.10. Processors and controllers fail at rate $1/2000$, disks fail at different rates from one cluster to another. These rates are $1/6000$, $1/8000$, $1/10000$, $1/12000$, $1/14000$, and $1/16000$. All components can fail in two modes with equal probability. The repair rate is 1 for one mode and 0.5 for the other. Components are repaired by a single repairman who chooses components at random from the set of failed components. Unfailed components continue to fail when the system is down. The second example is a modification of the first in which the number of controllers in each set is increased to 3 and the number of disks in each cluster is increased to 5, without any other aspect being modified.

The first example will be used to illustrate the method. It has $N = 36$ components and 10 component types: PA (processor A), PB (processor B), C1 and C2 (controllers), and D1, D2, D3, D4, D5 and D6 (disks), where disks D1, D2 and D3 are controlled by controllers C1, and disks D4, D5 and D6 are controlled by controllers C2. The structure function of the system is:

$$(\text{PA}[1] \vee \text{PB}[1]) \wedge \text{C1}[1] \wedge \text{C2}[1] \wedge \text{D1}[3] \wedge \text{D2}[3] \wedge \text{D3}[3] \wedge \text{D4}[3] \wedge \text{D5}[3] \wedge \text{D6}[3]\,,$$

where $c[n]$ are atoms which evaluate to true if at least n components of type c are unfailed. Table 1 gives the minimal cuts, where $c_1[n_1] \ldots c_k[n_k]$ denotes the bag including n_1 components of type c_1, ..., and n_k components of type c_k. The redundancy level is $L = \min_{m \in MC} |m| = 2$. Table 2 gives the failure bags of the model and, for each failure bag e, its importance $\text{Imp}(e)$, its activity $\text{Act}(e)$, and the upper bound rate $\lambda_{\text{ub}}(e)$. $FC = \{1, 2\}$ and the bounds f_1, f_2, and $g(k)$ are:

$$f_1 = \sum_{e \in E - \{e_3\}} \lambda_{\text{ub}}(e) = 8.436 \times 10^{-3}\,,$$

Table 1: Minimal cuts of the first example.

	description	cardinality
m_1	C1[2]	2
m_2	C2[2]	2
m_3	D1[2]	2
m_4	D2[2]	2
m_5	D3[2]	2
m_6	D4[2]	2
m_7	D5[2]	2
m_8	D6[2]	2
m_9	PA[4] PB[4]	8

Table 2: Failure bags of the first example and, for each failure bag e, Imp(e), Act(e) and $\lambda_{\rm ub}(e)$.

	description	$I(e)$	$A(e)$	$\lambda_{\rm ub}(e)$
e_1	PA[1]	7	1	2×10^{-3}
e_2	PB[1]	7	1	2×10^{-3}
e_3	PA[1] PB[1]	6	2	5×10^{-5}
e_4	C1[1]	1	1	10^{-3}
e_5	C2[1]	1	1	10^{-3}
e_6	D1[1]	1	1	6.667×10^{-4}
e_7	D2[1]	1	1	5×10^{-4}
e_8	D3[1]	1	1	4×10^{-4}
e_9	D4[1]	1	1	3.333×10^{-4}
e_{10}	D5[1]	1	1	2.857×10^{-4}
e_{11}	D6[1]	1	1	2.5×10^{-4}

$$f_2 = \lambda_{\rm ub}(e_3) = 5 \times 10^{-5}\,,$$
$$g(k) = 0.5, \quad F+1 \leq K \leq N\,.$$

The upper bound failure rate structures $f_{ij}(k,d)$ of the model for $F=1$ are shown in Fig. 10, where $f_{ij}(k,d)$ labels an arc going from node (k,d) to node $(k+i, d-j)$.

Our method can be transformed into the method proposed in [23] by making $C(k,d) = T(k)$. Thus, comparison of $C(k,d)$ with $T(k)$ indicates the potential for improvement of our method. Table 3 gives $T(k)$, $C(k)$ and $C(k,d)$ for the first example and $F=1$. In this case, $F+1 = L$ and (21), (16) $C(k) = T(k)$. After the iterative improvement algorithm $C(k,0) = C(k)$, i.e. the bounds $C(k,0)$ have not been improved in relation to their initial values. The values of $C(k,d)$, for $d > 0$ are however much smaller than $C(k) = T(k)$ and decrease as d increases. If a significant portion of the exits from G are made through states with failure distance > 0, the upper bounds $[C_{U,s}]_{\rm ub}$

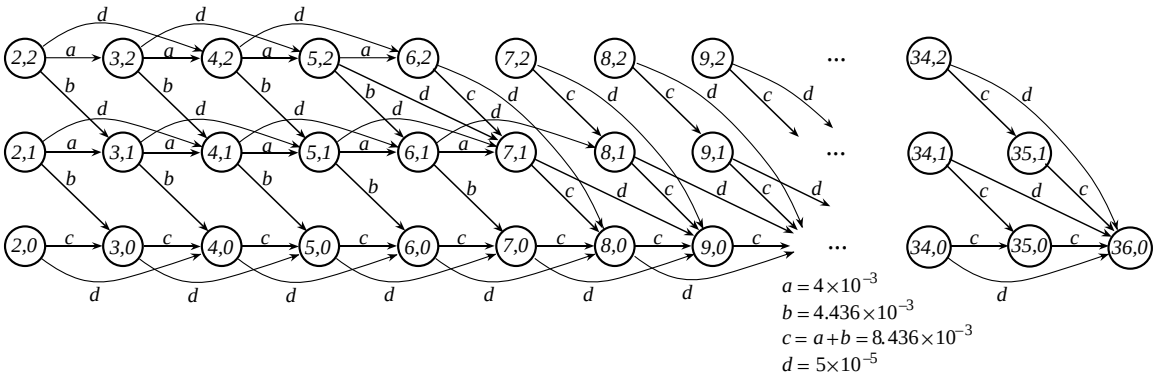

Figure 10: Upper bound failure rate structures $f_{ij}(k,d)$ for the first example and $F = 1$.

Table 3: First bounds $T(k)$, $C(k)$ and $C(k,d)$ for the first example and $F = 1$.

$T(2) = 2.0203$	$C(2) = 2.0203$	$C(2,2) = 6.3362 \times 10^{-4}$ $C(2,1) = 3.1980 \times 10^{-2}$ $C(2,0) = 2.0203$
$T(3) = 4.0407$	$C(3) = 4.0407$	$C(3,2) = 1.7756 \times 10^{-3}$ $C(3,1) = 8.0000 \times 10^{-2}$ $C(3,0) = 4.0407$
$T(4) = 6.0610$	$C(4) = 6.0610$	$C(4,2) = 3.5547 \times 10^{-3}$ $C(4,1) = 0.14409$ $C(4,0) = 6.0610$
$T(5) = 8.0814$	$C(5) = 8.0814$	$C(5,2) = 6.1603 \times 10^{-3}$ $C(5,1) = 0.22430$ $C(5,0) = 8.0814$

can be significantly smaller than $[T_{U,s}]_{\rm ub}$ and the upper bounds $[UA_s]_{\rm ub}$ can be significantly tighter than $[UA_s]'_{\rm ub}$. In other words, our method will improve significantly the method described in [23] if down states are relatively rare in the frontier of the non-generated state space U.

The iterative improvement algorithm of the bounds $C(k,d)$ has very fast convergence. Typically, between 5 and 10 improvement steps are enough to achieve convergence in all $C(k,d)$ values up to the seventh significant digit. As an illustration, Table 4 gives the evolution of the bounds $C(2,d)$ for the first example and when $F = 1$.

We first study the impact of F on the tightness of the bounds obtained by our method. Table 5 gives the bounds obtained for the first example for several values of K, with F ranging from 0 to K. Table 6 shows the corresponding results for the second example. We can see that the tightness of the bounds improves when F is increased from 0 to 1 for the first example and from 0 to 2 for the second example, but deteriorates with further increase in F. In both cases, $F = L - 1$ gives the tighter bounds. This behavior is in contrast with the behavior of the method proposed in [23], which

Table 4: Evolution of the bounds $C(2,d)$ in the iterative improvement algorithm for the first example and $F=1$.

step	$C(2,2)$	$C(2,1)$	$C(2,0)$
0	2.0203	2.0203	2.0203
1	4.0493×10^{-2}	4.0493×10^{-2}	2.0203
2	1.0165×10^{-3}	3.2026×10^{-2}	2.0203
3	6.3697×10^{-4}	3.1981×10^{-2}	2.0203
4	6.3365×10^{-4}	3.1980×10^{-2}	2.0203
5	6.3362×10^{-4}	3.1980×10^{-2}	2.0203
6	6.3362×10^{-4}	3.1980×10^{-2}	2.0203

gives bounds which always improve when F is increased and, therefore, gives tightest bounds for $F=K$. This behavior can be explained as follows. With $F=L$ some of the return states are down, and the corresponding $C_{G,s}$ are relatively large, since the return state s is visited at least once by Y_G^s. The selection $F=L-1$ gives $C_{G,s}$ which are smaller in the worst case and gives tighter upper bounds. We have however no proof that $F=L-1$ is always the best option. Then, a reasonable possibility is to obtain bounds for F varying from 0 to K and keep the better bounds. The algorithm given in Fig. 7 can be modified easily to achieve this, by computing $T(k)$ and $C(k,d)$ and solving 5 linear systems of size $|G|$ for each value of F. Since most of the computational effort of our method for given F is spent in generating the detailed model, the extra computational cost would be moderate.

Table 7 gives the unavailability bounds obtained for the first example by the method proposed in [23] with $F=K$ and our method for the optimum F for the two first examples and several values of K. Besides the lower and upper unavailability bounds and the unavailability band, we also show the improvement of our method, defined as the factor by which the unavailability band is reduced in relation to [23]. We also give the cardinality of the generated state space G. Our method always gives tighter bounds. The improvement factor is greater for the second example. This can be attributed to many fewer down states and greater failure distances in the frontier of G.

We next illustrate how the tightness of the bounds given by our method is degraded when only minimal cuts up to a given cardinality M are known. As pointed out in Section 5, limiting the cardinality of the minimal cuts considered reduces significantly the associated overheads when the number of minimal cuts is very large (of the order of tens of thousands). Fig. 11 plots the relative unavailability band $([UA]_{\rm ub}-[UA]_{\rm lb})/[UA]_{\rm lb}$ achieved for the first example with $K=2$ and 3, and the second example with $K=3$ and 4, using the optimum F in all cases, as a function of M. We consider increasing values of M until there is no apparent degradation due to limited knowledge of minimal cuts. The band obtained with the method described in [23] corresponds to $M=0$. Three regions are very clearly noticed in the curves. The first one corresponds to values $0 \le M \le K$, where all non-generated states reachable from G through failure transitions have more

Table 5: Dependence of the bounds obtained by our method on F for the first example. We give, in this order, the lower unavailability bound, the upper unavailability bound, and the unavailability band.

	$K = 2$	$K = 3$	$K = 4$
$F = 0$	3.2313×10^{-6} 3.4746×10^{-6} 2.4322×10^{-7}	3.3167×10^{-6} 3.3239×10^{-6} 7.1676×10^{-9}	3.3192×10^{-6} 3.3194×10^{-6} 1.6469×10^{-10}
$F = 1$	3.2313×10^{-6} 3.4627×10^{-6} 2.3133×10^{-7}	3.3167×10^{-6} 3.3237×10^{-6} 7.0360×10^{-9}	3.3192×10^{-6} 3.3194×10^{-6} 1.6254×10^{-10}
$F = 2$	3.2313×10^{-6} 5.2557×10^{-6} 2.0244×10^{-6}	3.3167×10^{-6} 3.3465×10^{-6} 2.9755×10^{-8}	3.3192×10^{-6} 3.3196×10^{-6} 4.2746×10^{-10}
$F = 3$		3.3167×10^{-6} 3.3694×10^{-6} 5.2746×10^{-8}	3.3192×10^{-6} 3.3199×10^{-6} 6.9468×10^{-10}
$F = 4$			3.3192×10^{-6} 3.3202×10^{-6} 9.6539×10^{-10}

than M failed components and, therefore, are assumed down. In this region an increase of M gives small improvements in the bounds. Substantial improvement in the bounds is achieved when M is increased beyond K. In this region, a substantial portion of the non-generated states reachable from G through failure transitions have failure distance > 0 and give contributions to $[C_{U,s}]_{\mathrm{ub}}$ which are much smaller than previously. The region in which substantial improvement occurs extends typically to $M = K + 2$. Beyond that point, the bounds do not improve substantially. The results indicate that significant improvements over the bounds given by the method described in [23] can be obtained when $M > K$, and that the improvement obtained by increasing M decreases as M increases.

We also obtain results for a third example with a substantially more complex structure function than the two previous examples. The example is a fault-tolerant distributed real-time system. The architecture of the system and its configuration for a particular pattern of failed components are shown in Fig. 12. A dual configuration of data processing units (DPUs) commands control subsystems located at remote sites. Each control subsystem comprises two redundant control units (CUs) working in hot-standby redundancy. The system can be accessed through two redundant front-ends connected to the DPUs. The DPUs and CUs communicate using a redundant local area network (LAN) to which each DPU and each CU has access through dedicated communication processors (CPs). Components fail at constant rates λ_{FE}, λ_{DPU}, λ_{CU}, λ_{CP}, and λ_{L}. Two failed modes are considered for the DPUs: “soft” and “hard”. The first mode occurs with probability α and can be

Table 6: Dependence of the bounds obtained by our method on F for the second example (we give, in this order, the lower unavailability bound, the upper unavailability bound, and the unavailability band).

	$K = 3$	$K = 4$	$K = 5$
$F = 0$	4.5418×10^{-9} 5.3420×10^{-9} 8.0016×10^{-10}	4.7207×10^{-9} 4.7498×10^{-9} 2.9057×10^{-11}	4.7268×10^{-9} 4.7276×10^{-9} 8.3150×10^{-13}
$F = 1$	4.5418×10^{-9} 5.3380×10^{-9} 7.9612×10^{-10}	4.7207×10^{-9} 4.7497×10^{-9} 2.8992×10^{-11}	4.7268×10^{-9} 4.7276×10^{-9} 8.3053×10^{-13}
$F = 2$	4.5418×10^{-9} 5.2951×10^{-9} 7.5325×10^{-10}	4.7207×10^{-9} 4.7487×10^{-9} 2.7941×10^{-11}	4.7268×10^{-9} 4.7276×10^{-9} 8.0948×10^{-13}
$F = 3$	4.5418×10^{-9} 4.5234×10^{-8} 4.0693×10^{-8}	4.7207×10^{-9} 5.2878×10^{-9} 5.6709×10^{-10}	4.7268×10^{-9} 4.7349×10^{-9} 8.1446×10^{-12}
$F = 4$		4.7207×10^{-9} 5.8329×10^{-9} 1.1122×10^{-9}	4.7268×10^{-9} 4.7423×10^{-9} 1.5514×10^{-11}
$F = 5$			4.7268×10^{-9} 4.7497×10^{-9} 2.2962×10^{-11}

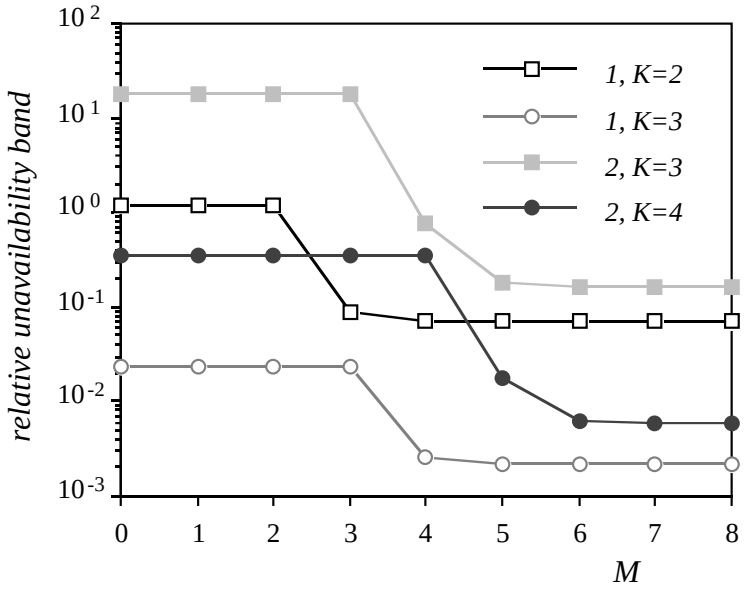

Figure 11: Relative band achieved for the first (1) and second (2) examples for several values of K and optimum F when only minimal cuts up to cardinality M are known (the band obtained with the method proposed in [23] corresponds to $M = 0$).

Table 7: Comparison of the method described in [23] with $F = K$ with the method proposed here with optimum F for the two first examples (for each method, we give, in this order, the lower unavailability bound, the upper unavailability bound, and the unavailability band).

example	K	[23]	proposed	improvement
1	2	3.2313×10^{-6}	3.2313×10^{-6}	16.6
	$\|G\| = 231$	7.0744×10^{-6}	3.4627×10^{-6}	
		3.8430×10^{-6}	2.3133×10^{-7}	
1	3	3.3167×10^{-6}	3.3167×10^{-6}	10.8
	$\|G\| = 1763$	3.3927×10^{-6}	3.3237×10^{-6}	
		7.6022×10^{-8}	7.0360×10^{-9}	
1	4	3.3192×10^{-6}	3.3192×10^{-6}	7.63
	$\|G\| = 10464$	3.3204×10^{-6}	3.3194×10^{-6}	
		1.2400×10^{-9}	1.6254×10^{-10}	
2	3	4.5418×10^{-9}	4.5418×10^{-9}	107
	$\|G\| = 1771$	8.5492×10^{-8}	5.2951×10^{-9}	
		8.0950×10^{-8}	7.5325×10^{-10}	
2	4	4.7207×10^{-9}	4.7207×10^{-9}	59.5
	$\|G\| = 10616$	6.3828×10^{-9}	4.7487×10^{-9}	
		1.6621×10^{-9}	2.7941×10^{-11}	
2	5	4.7268×10^{-9}	4.7268×10^{-9}	37.7
	$\|G\| = 52916$	4.7573×10^{-9}	4.7276×10^{-9}	
		3.0480×10^{-11}	8.0948×10^{-13}	

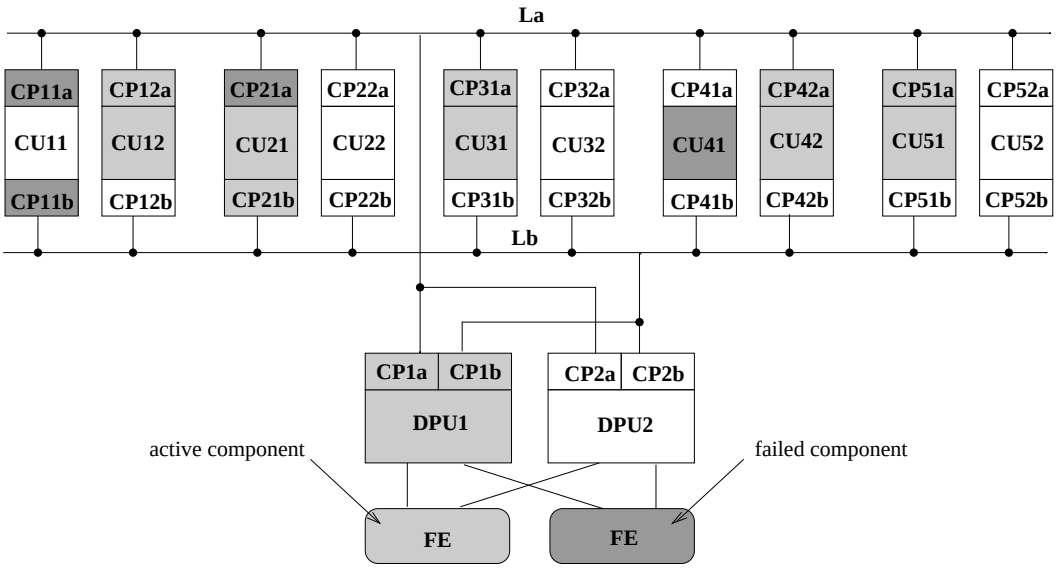

Figure 12: Architecture of the fault-tolerant distributed real-time system and a configuration of the system.

recovered by an operator restart; the second mode occurs with probability $1 - \alpha$ and requires hardware repair. Coverage is assumed perfect for all faults except those of the DPUs, which take the system down with a probability $1 - C$. Lack of coverage is modeled by propagating the failure of one DPU to the other DPU. There are three repair teams. The first repairs LANs and CPs, with preemptive priority given to LANs. The second repairs FEs, CUs and DPUs in "hard" failed mode, with preemptive priority given first to DPUs, followed by FEs and then CUs. The third makes DPU restarts. Each team includes only one repairman. Failed components with the same priority are taken at random for repair. The repair rates are denoted by μ_{FE}, μ_{DPUh}, μ_{DPUs}, μ_{CU}, μ_{CP}, and μ_{L}. The system is operational if one unfailed DPU can communicate with at least one unfailed CU of each control subsystem. Different LANs can be used for communication between the active DPU and the active CU of each control subsystem, but the communication between each pair has to be direct, i.e. involving only one CP from each unit and one LAN. Fig. 12 illustrates the configurations which the system can adopt. Each such configuration includes an active DPU, an active CU and an active CP associated to the CU at each site. Depending on whether one or both LANs are used, the active DPU will have one or both CPs associated with it active. The front-ends can be conceptualized as instances of the same component type. However, the interconnection relationships make it mandatory to consider all other components as unique representatives of different component types. The resulting CTMC has about 4.6×10^{11} states. We use the sets of model parameter values given in Table 8. These sets are chosen to represent different scenarios. The values for failure and repair rates chosen for the sets a, b and c are meant to be typical, i.e. repair rate/failure rate ratios of four to five orders of magnitude. These sets only differ in the value chosen for C, the coverage to DPU failures. In set a, coverage failures are the dominant source of system failures; in set c, resource exhaustion is the dominant source; and in set b, both are important. Set d is obtained from set b by making all repairs 10 times slower; set e is obtained from set c by making failure rates 10 times slower.

Table 8: Sets of model parameter values for the fault-tolerant distributed real-time system.

set	a	b	c	d	e
λ_{FE}	2×10^{-4}	2×10^{-4}	2×10^{-4}	2×10^{-4}	2×10^{-5}
λ_{DPU}	10^{-4}	10^{-4}	10^{-4}	10^{-4}	10^{-5}
λ_{CU}	10^{-4}	10^{-4}	10^{-4}	10^{-4}	10^{-5}
λ_{L}	10^{-5}	10^{-5}	10^{-5}	10^{-5}	10^{-6}
λ_{CP}	5×10^{-5}	5×10^{-5}	5×10^{-5}	5×10^{-5}	5×10^{-6}
α	0.9	0.9	0.9	0.9	0.9
C	0.99	0.999	0.9999	0.999	0.9999
μ_{FE}	1	1	1	0.1	1
μ_{DPUh}	1	1	1	0.1	1
μ_{DPUs}	4	4	4	0.4	4
μ_{CU}	1	1	1	0.1	1
μ_{L}	0.5	0.5	0.5	0.05	0.5
μ_{CP}	1	1	1	0.1	1

Table 9 gives the results obtained with the method proposed in [23] without state cloning ($F = K$) and our method with optimum F for the five model parameter sets considered for the fault-tolerant distributed real-time example. We increase K until our method gives approximately two digits of accuracy, starting with $K = 2$ to have some down state in G and a lower unavailability bound > 0. We can note that in both methods the bounds are tighter as the repair rates of the model are comparatively faster in relation to the failure rates. However, comparison of the results for sets a, b and c and $K = 2$ shows that the improvement achieved by our method is adversely affected by an increased importance of coverage failures. These failures can introduce both short paths to down states with significantly higher probabilities than the paths associated with resource exhaustion and dominant contributions to the band which offset the clear reduction achieved by our method in the contributions to the band associated with transitions from states in G to operational states in U. Anyhow, the increased accuracy of our method is once more significant. Thus, for instance, to achieve the desired accuracy (two digits) using the method proposed in [23], we will need $K = 3$ instead of $K = 2$ for sets b and c, and $K = 4$ instead of $K = 3$ for set d, with a significant increase in the number of detailed states (the number of detailed states is 112,050 for $K = 4$).

The structure function of the third example has 512 minimal cuts: 8 of cardinality 2, 48 of cardinality 3, 96 of cardinality 4 and 360 of cardinality 6. Thus, it is a good example for analyzing the overhead due to the computation of failure distances. The storage overhead is only significant when the number of minimal cuts is comparable to the number of generated states. Thus, we will concentrate on the time overhead. We profiled the code for case c, $K = 3$ and $F = 0$, taking for the parameter R of the algorithms given in Figs 5 and 6 the value 2. The bounds, including model generation, were obtained in 27.98 s in a SPARC10 workstation. There was an average of

Table 9: Comparison of the method described in [23] with $F = K$ with the method proposed here with optimum F for the fault-tolerant distributed real-time system (for each case, we give, in this order, the lower unavailability bound, the upper unavailability bound, and the unavailability band).

set	K	[23]	proposed	improvement
a	2	6.6983×10^{-7}	6.6983×10^{-7}	2.58
	$\|G\| = 861$	7.5155×10^{-7}	7.0147×10^{-7}	
		8.1720×10^{-8}	3.1641×10^{-8}	
a	3	6.7277×10^{-7}	6.7277×10^{-7}	1.37
	$\|G\| = 11483$	6.7291×10^{-7}	6.7287×10^{-7}	
		1.4297×10^{-10}	1.0462×10^{-10}	
b	2	2.3031×10^{-7}	2.3031×10^{-7}	7.35
	$\|G\| = 861$	2.8832×10^{-7}	2.3820×10^{-7}	
		5.8018×10^{-8}	7.8912×10^{-9}	
c	2	1.8636×10^{-7}	1.8636×10^{-7}	10.1
	$\|G\| = 861$	2.4200×10^{-7}	1.9187×10^{-7}	
		5.5648×10^{-8}	5.5163×10^{-9}	
d	2	1.7747×10^{-5}	1.7748×10^{-5}	4.84
	$\|G\| = 861$	7.3899×10^{-5}	2.9353×10^{-5}	
		5.6151×10^{-5}	1.1606×10^{-5}	
d	3	1.9036×10^{-5}	1.9036×10^{-5}	2.18
	$\|G\| = 11483$	1.9943×10^{-5}	1.9452×10^{-5}	
		9.0733×10^{-7}	4.1579×10^{-7}	
e	2	2.3136×10^{-9}	2.3136×10^{-9}	7.96
	$\|G\| = 861$	2.3715×10^{-9}	2.3208×10^{-9}	
		5.7977×10^{-11}	7.2874×10^{-12}	

about 11 minimal cut touches per state in the frontier (11,289 out of 11,483 were frontier states). This is a very small number considering that the model has 512 minimal cuts and 40 failure bags (and, thus, the trivial procedure based on (17) would involve 20,480 touches per generated state), and illustrates the efficiency of the techniques described in Section 4.1. The time overhead due to failure distance computation was 13.7%, but the overhead that depends on the number of minimal cuts touched was only 0.28%. Thus, we feel that the time overhead will remain of the order of 10% even when the number of minimal cuts is much larger. Computation of all minimal cuts using the algorithm described in [7] took 1.74 s.

7 Conclusions

We have developed a new method to bound steady-availability which exploits the concept of failure distance. We have proved that the method gives bounds which are guaranteed to be not worse than the bounds achieved by a previous bounding method. Numerical experiments have shown that the improvement in tightness can be significant, especially for systems with high redundancy. We have shown that the overheads in time and storage of our method are small and are well paid off by the improved tightness. It remains to be seen whether the bounds proposed here can still be improved at a moderate computational overhead. The application of the concepts developed here to obtain tight bounds for other dependability and performability measures and their derivatives, as required for sensitivity analysis, can be undertaken in the future.

Appendix A. Proof of Theorem 3

Without loss of generality, assume that the transient states of Y are sorted following the subset ordering $B_1, B_2, \ldots, B_n$. For notational conciseness, let $\tau_i = \tau(i, Y)$ and $\tau'_k = \tau(B_k, Y) = \sum_{i \in B_k} \tau_i$. Note that $\tau'_k > 0$. Let the vectors $\boldsymbol{\tau} = (\tau_i)_{i \in B}$, $\boldsymbol{\pi} = (\pi_i)_{i \in B}$ and let $\mathbf{A}$ be the transition rate matrix of Y restricted to B. $\boldsymbol{\tau}$ satisfies the linear system

$$\boldsymbol{\tau}^T \mathbf{A} = -\boldsymbol{\pi}^T \,. \tag{63}$$

Let $w_i^k = \tau_i / \tau'_k$, $i \in B_k$, $1 \leq k \leq n$. Note that $w_i^k > 0$ and $\sum_{i \in B_k} w_i^k = 1$. Defining the column vectors $\mathbf{w}(k) = (w_i^k)_{i \in B_k}$, $\boldsymbol{\pi}(k) = (\pi_i)_{i \in B_k}$, we can rewrite (63) as

$$\Big(\tau'_1 \mathbf{w}(1)^T \cdots \tau'_n \mathbf{w}(n)^T\Big) \begin{pmatrix} \mathbf{A}_{11} & \cdots & \mathbf{A}_{1n} \\ & \ddots & \\ \mathbf{A}_{n1} & \cdots & \mathbf{A}_{nn} \end{pmatrix} = -\Big(\boldsymbol{\pi}(1)^T \cdots \boldsymbol{\pi}(n)^T\Big),$$

where $\mathbf{A}_{kl}$ are the blocks of $\mathbf{A}$ induced by the partition of B. This block decomposition gives the set of equations:

$$\sum_{k=1}^{n} \tau'_k \mathbf{w}(k)^T \mathbf{A}_{kl} = -\boldsymbol{\pi}(l)^T \,, \;\; 1 \leq l \leq n \,.$$

Postmultiplying by **1**, a column vector of all ones with appropriate dimension:

$$\sum_{k=1}^{n} \tau'_k \mathbf{w}(k)^T \mathbf{A}_{kl} \mathbf{1} = -\boldsymbol{\pi}(l)^T \mathbf{1}\,, \;\; 1 \le l \le n\,.$$

Defining $\pi'_k = \boldsymbol{\pi}(k)^T \mathbf{1} = \sum_{i \in B_k} \pi_i$, $\lambda'_{b_k,b_l} = \mathbf{w}(k)^T \mathbf{A}_{kl} \mathbf{1} = \sum_{i \in B_k} w_i^k \lambda_{i,B_l}$, $k \neq l$, and $\lambda'_{b_k} = -\mathbf{w}(k)^T \mathbf{A}_{kk} \mathbf{1}$, we get

$$\sum_{\substack{k=1 \\ k \neq l}}^{n} \tau'_k \lambda'_{b_k,b_l} - \tau'_l \lambda'_{b_l} = -\pi'_l, \;\; 1 \le l \le n\,.$$

Thus, $\boldsymbol{\tau}' = (\tau'_k)_{1 \le k \le n}$ satisfies the linear system

$$\boldsymbol{\tau}'^T \mathbf{A}' = -\boldsymbol{\pi}'^T\,,$$

with $\boldsymbol{\pi}' = (\pi'_k)_{1 \le k \le n}$ and

$$\mathbf{A}' = \begin{pmatrix} -\lambda'_{b_1} & \lambda'_{b_1,b_2} & \cdots & \lambda'_{b_1,b_n} \\ \lambda'_{b_2,b_1} & -\lambda'_{b_2} & \cdots & \lambda'_{b_2,b_n} \\ & \cdots & & \\ \lambda'_{b_n,b_1} & \lambda'_{b_n,b_2} & \cdots & -\lambda'_{b_n} \end{pmatrix}. \tag{64}$$

In summary, under the condition $\lambda'_{b_k,a} = \lambda'_{b_k} - \sum_{\substack{l=1 \\ l \neq k}}^{n} \lambda'_{b_k,b_l} \ge 0$, $1 \le k \le N$, $\tau'_k = \tau(B_k, Y)$ ($< \infty$ since all states in B of Y are transient) is the mean time to absorption in state b_k of the transient CTMC Y' with state space $\{b_1, b_2, \ldots, b_N\} \cup \{a\}$, transition rate matrix (64), and initial probability distribution $P[Y'(0) = b_k] = \pi'_k$, $1 \le k \le N$. The transition rates λ'_{b_k,b_l} satisfy the conditions of the theorem. It remains to be seen whether the transition rates to the absorbing state $\lambda'_{b_k,a}$ also satisfy those conditions and are ≥ 0. First, note that the output rates of Y' can be written as

$$\lambda'_{b_k} = -\mathbf{w}(k)^T \mathbf{A}_{kk} \mathbf{1} = \sum_{i \in B_k} w_i^k \lambda_i - \sum_{i \in B_k} w_i^k \lambda_{i,B_k}\,.$$

Then, using $\lambda'_{b_k,a} = \lambda'_{b_k} - \sum_{\substack{l=1 \\ l \neq k}}^{n} \lambda'_{b_k,b_l}$ and $\lambda_{ia} = \lambda_i - \sum_{l=1}^{n} \lambda_{i,B_l}$:

$$\begin{aligned} \lambda'_{b_k,a} &= \lambda'_{b_k} - \sum_{\substack{l=1 \\ l \neq k}}^{n} \lambda'_{b_k,b_l} = \sum_{i \in B_k} w_i^k \lambda_i - \sum_{i \in B_k} w_i^k \lambda_{i,B_k} - \sum_{\substack{l=1 \\ l \neq k}}^{n} \sum_{i \in B_k} w_i^k \lambda_{i,B_l} \\ &= \sum_{i \in B_k} w_i^k \Big(\lambda_i - \sum_{l=1}^{n} \lambda_{i,B_l} \Big) = \sum_{i \in B_k} w_i^k \lambda_{ia} \ge 0\,. \quad \square \end{aligned}$$

References

[1] R.E. Barlow and F. Proschan, *Statistical Theory of Reliability and Life Testing. Probability Models*, McArdle Press, Silver Spring, 1981.

[2] C. Béounes, M. Aguéra, J. Arlat, S. Bachman, C. Bourdeau, J. E. Doucet, K. Kanoun, J. C. Laprie, S. Metge. J. Moreira de Souza, D. Powell and P. Spiesser, "SURF-2: A Program for Dependability Evaluation of Complex Hardware and Software Systems," *Proc. 23rd IEEE Int. Symp. on Fault-Tolerant Computing (FTCS-23)*, Toulouse, 1993, pp. 668–673.

[3] U. N. Bhat, *Elements of Applied Stochastic Processes*, 2nd edition, John Wiley and Sons, 1984.

[4] J. A. Carrasco, "Failure distance-based Simulation of Repairable Fault-Tolerant Systems," *Computer Performance Evaluation. Modeling Techniques and Tools*, Elsevier, 1992, pp. 351–365.

[5] J. A. Carrasco, A. Calderón and J. Escribá, "Two New Algorithms to Compute Steady-state Bounds for Markov Models with Slow Forward and Fast Backward Transitions," *Proc. 4th Int. Workshop on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems*, February 1996, pp. 89–95.

[6] J.A. Carrasco and J. Figueras, "METFAC: Design and Implementation of a Software Tool for Modeling and Evaluation of Complex Fault-Tolerant Computing Systems," *Proc. of the 16th Int. Symp. on Fault-Tolerant Computing FTCS-16*, 1986, pp. 424–429.

[7] J. A. Carrasco and V. Suñé, "An Algorithm to Find Minimal Cuts of s-Coherent Fault Trees with Event classes using a Decision Tree," *Proc. XII Conference on Design of Circuits and Integrated Systems*, Sevilla, Spain, November 1997, pp. 279–284.

[8] G. Chiola. C. Dutheillet, G. Franceschinis and S. Haddad, "Stochastic Well-Formed Colored Nets and Symmetric Modeling Applications," *IEEE Trans. on Computers*, vol. 42, no. 11, November 1993, pp. 1343–1360.

[9] G. Ciardo, J. Muppala and K. Trivedi, "SPNP: Stochastic Petri net Package," *Proc. 3rd IEEE Int. Workshop on Petri Nets and Performance Models (PNPM89)*, Kyoto, 1989, pp. 142–150.

[10] E. Çinlar, *Introduction to Stochastic Processes*, Prentice-Hall, 1975.

[11] P.J. Courtois and P. Semal, "Bounds for the positive eigenvectors of nonnegative matrices and for their approximations," *J. of the ACM*, vol. 31, no. 4, pp. 804–825, October 1984.

[12] P.J. Courtois and P. Semal, "Computable bounds for conditional steady-state probabilities in large Markov chains and queueing models," *IEEE J. Select. Areas Commun.*, vol. SAC-4, no. 6, pp. 926–937, September 1986.

[13] J. Couvillion, R. Freire, R. Johnson, W. Obal II, A. Qureshi, M. Rai, W. Sanders and J. Tvedt, "Performability modelling with UltraSAN," *IEEE Software*, September 1991, pp. 69–80.

[14] M. Dal Cin, "Availability Analysis of a Fault-Tolerant Computer System," *IEEE Trans. on Reliability*, vol. R-29, no. 3, August 1980, pp. 265–268.

[15] A. Goyal, W.C. Carter, E. de Souza e Silva, S.S. Lavenberg and K.S. Trivedi, "The System Availability Estimator," *Proc. of the 16th Int. Symp on Fault-Tolerant Computing FTCS-16*, 1986, pp. 84–89.

[16] A. Goyal, P. Shahabuddin, P. Heidelberger, V.F. Nicola and P.W. Glynn, "A Unified Framework for Simulating Markovian Models of Highly Dependable Systems," *IEEE Trans. on Computers*, vol. 41, no. 1, pp. 36–51, January 1992.

[17] P. Heidelberger, J. K. Muppala and K. S. Trivedi, "Accelerating Mean Time to Failure Computations," IBM Research Report RC 20415, March 1996.

[18] J.G. Kemeny and J.L. Snell, *Finite Markov Chains*, 2nd edition, Springer-Verlag, New York, 1978.

[19] W.S. Lee, D.L. Grosh, F.A. Tillman and C.H. Lie, " Fault Tree Analysis, Methods and Applications –A Review," *IEEE Trans. on Reliability*, vol. R-34, no. 3, August 1985, pp. 194–203.

[20] J.C.S. Lui and R. Muntz, “Evaluating Bounds on Steady-State Availability of Repairable Systems from Markov Models,” in *Numerical Solution of Markov chains*, Marcel Dekker, New York, pp. 435–454, 1991.

[21] J.C.S. Lui and R.R. Muntz, “Computing Bounds on Steady State Availability of Repairable Computer Systems,” *Journal of the ACM*, vol. 41, no. 4, July 1994, pp. 676–707.

[22] S.V. Makam and A. Avizienis, “ARIES 81: A reliability and life-cycle evaluation tool for fault-tolerant systems,” in *Proc. 12th Int. Symp. on Fault-Tolerant Computing FTCS-12*, June 1982, pp. 266–274.

[23] R.R. Muntz, E. de Souza e Silva and A. Goyal, “Bounding Availability of Repairable Computer Systems,” *IEEE Trans. on Computers*, vol. 38, no. 12, pp. 1714–1723, December 1989.

[24] D.M. Rasmuson and N.H. Marshall, “ FATRAM-A Core Efficient Cut-Set Algorithm,” *IEEE Trans. on Reliability*, vol. R-27, no. 4, October 1978, pp. 250–253.

[25] R.A. Sahner and K.S. Trivedi, “Reliability Modeling Using SHARPE,” *IEEE Trans. on Reliability*, vol. R-36, no. 2, pp. 186–193, June 1987.

[26] E. de Souza e Silva and P.M. Ochoa, “State Space Exploration in Markov Models,” *Performance Evaluation Review*, vol. 20, no. 1, June 1992, pp. 152–166.

[27] K.S. Trivedi, J.B. Dugan, R.R. Geist and M.K. Smotherman, “Hybrid reliability modeling of fault-tolerant computer systems,” *Comput. Elec. Eng.*, vol. 11, pp. 87–108, 1984.

[28] R. S. Varga, *Matrix Iterative Analysis*, Prentice-Hall, 1962.

[29] M. Veeraraghavan and K.S. Trivedi, “A Combinatorial Algorithm for Performance and Reliability Analysis Using Multistate Models,” *IEEE Trans. on Computers*, vol. 43, no. 2, pp. 229–234, February 1994.